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

Интеллектуальные Системы.

Теория и приложения

ТОМ **29** ВЫПУСК 3 * **2025**

МОСКВА

УДК 519.95; 007:159.955 ISSN 2411–4448
ББК 32.81 Издаётся с 1996 г.

В журнале «Интеллектуальные системы. Теория и приложения» публикуются научные достижения в области теории и приложений интеллектуальных систем, новых информационных технологий и компьютерных наук.
Издание журнала осуществляется под эгидой МГУ имени М.В. Ломоносова, Научного Совета по комплексной проблеме «Кибернетика» РАН, Отделения «Математическое моделирование технологических процессов» МАТН.
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. Входит в дополнительный перечень научных изданий, в которых могут быть опубликованы результаты диссертаций, защищаемых в МГУ.

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@intsysmagazine.ru

*) Прежнее название журнала: «Интеллектуальные системы».

ОГЛАВЛЕНИЕ

Часть 1
Общие проблемы теории интеллектуальных систем

Введение в логические процессы. Общая схема функционирования решателя

А. С. Подколзин[1]

В статье описывается общая схема функционирования решателя математических задач. Рассказывается как происходит сканирование задачи, как запускать решение задачи и как осуществлять пошаговый просмотр. Приводится большое количество упражнений по вводу и решению задач.

Ключевые слова: решатель математических задач, логические процессы, логический язык, логическая формализация задач.

1. Введение

Данная статья является второй в цикле статей, посвященных практикуму по решателю математических задач. Она является логическим продолжением статьи [1]. В данной статье мы опишем общую схему функционирования решателя математических задач и приведем упражнения по вводу и решению задач из разных разделов математики. Решатель и заложенные в него принципы подробно описаны в монографиях [2, 3, 4, 5, 6, 7, 8, 9, 10].

2. Сканирование задачи

При обычном программировании для решения задач заданного типа разрабатывается один общий алгоритм, который гарантирует получение за конечное число шагов ответа в случае любой конкретной задачи этого типа. Такой алгоритм представляется в виде нескольких взаимодействующих между собой блоков, обеспечивающих циклический процесс постепенного упрощения параметров решаемой задачи (в частности, уменьшения оставшегося объема перебора для процедур переборного типа) — вплоть до получения окончательного ответа. Каждый из блоков несет здесь вполне определенную функциональную нагрузку, как правило, основанную на том или ином теоретическом утверждении, связанном с обрабатываемыми алгоритмом объектами. Его можно рассматривать как прием для достижения определенной текущей подцели. В свою очередь,

[1] *Подколзин Александр Сергеевич* — д.ф.м.н., профессор каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: alexander.p@yandex.ru.

Podkolzin Alexander Sergeevich — Dr. of Sc., Professor, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

отдельный блок сам определяется схемой своих подблоков, и т.д. Здесь возникает иерархия уровней, в которой на каждом уровне несколько приемов (как правило, не более одного — двух десятков) оказываются связаны в вычислительный цикл, так что после выполнения очередного приема известно, к какому приему следует переходить дальше.

К сожалению, такого рода четкую программу действий удается составить лишь для достаточно узких классов задач. Хорошо известны результаты об алгоритмической неразрешимости различных общих проблем, и даже в случае наличия алгоритма он часто оказывается практически неприемлемым по своей трудоемкости. Для преодоления возникающих здесь трудностей приходится резко увеличивать количество используемых приемов, чтобы охватить алгоритмическими возможностями если не все многообразие задач предметной области, то хотя бы возможно больший класс задач, встречающихся в реальных ситуациях. Эти приемы создаются уже не путем теоретического проектирования алгоритма, а извлекаются из последовательности конкретных обучающих задач и складываются в одну общую базу приемов. Количество приемов, используемых для решения задач в предметной области, обычно существенно больше чем в алгоритмической процедуре указанного выше блочного типа - сотни и тысячи вместо десятков. Последовательность их работы заранее не фиксирована, и после срабатывания очередного приема необходим цикл поиска следующего применяемого приема. Разумеется, при увеличении числа приемов, работающих в общем процессе, существенно усложняется регулировка взаимодействия между ними, и возникает необходимость в длительной эмпирической оптимизации их решающих правил.

Для организации цикла поиска очередного применяемого приема в решателе используется процедура сканирования задачи, которую можно представлять как своего рода модель внутреннего логического зрения. Для большинства приемов активизация их при рассмотрении задачи начинается с обнаружения в логических структурах данных некоторого логического символа, заранее выбранного для этого приема. В качестве такого ключевого логического символа берется обычно некоторый логический символ, появление которого необходимо для возможности применения рассматриваемого приема, причем при нескольких возможных выборах предпочтение отдается наиболее редко встречающемуся логическому символу (для уменьшения потерь времени при попытках применения приема). Указанное закрепление за приемами логических символов предопределяет организацию всей базы приемов решателя по принципу энциклопедии: она распадается на группы приемов, “принадлежащих” соответствующим символам, причем каждая группа реализуется в виде отдельной алгорит-

мической процедуры A_f — программы логического символа f. В особых случаях прием не удается связать с каким-либо конкретным логическим символом, появление которого является необходимым для срабатывания. Такие приемы (их совсем немного) распределены по четырем логическим символам — названиям "доказать", "описать", "преобразовать", "исследовать" типов задач, при решении которых возможно их применение.

Текущая ситуация, возникающая в процессе работы решателя, описывается последовательностью задач $Z_1, Z_2, \ldots, Z_N$, где Z_{i+1} — вспомогательная задача, введенная при решении задачи Z_i $(i = 1, \ldots, N-1)$. В этой последовательности (далее называем ее цепью задач) задача Z_1 является фиктивной — она создается автоматически при запуске логической системы и используется для организации интерфейса. Эта задача (далее называемая исходной задачей) имеет тип "исследовать" и единственную однобуквенную посылку "вход". При просмотре посылок данной задачи решатель обращается к программе логического символа "вход", которая и является программой интерфейса логической системы. С помощью интерфейса можно ввести некоторую задачу Z_2, которая далее и решается, порождая вспомогательные задачи $Z_3, \ldots, Z_N$. Будем называть задачу Z_N текущей задачей; Z_2 — корневой задачей. Исходная задача, кроме организации интерфейса, выполняет функции "доски объявлений" — различные процедуры решателя могут обмениваться между собой сообщениями, размещая их в списке комментариев к посылкам этой задачи.

Каждая задача Z_i $(i = 1, \ldots, N)$ характеризуется натуральным числом M_i, называемым ее максимальным уровнем и определяющим уровень средств, отведенных для ее решения (по исчерпании этих средств выдается отказ), целым неотрицательным числом m_i, $m_i \leq M_i$, называемым текущим уровнем этой задачи и определяющим уровень средств, среди которых в текущий момент ведется поиск очередного преобразования задачи Z_i, а также вспомогательной информацией, необходимой для возобновления прерванной процедуры решения задачи Z_{i-1} по окончании решения задачи Z_i.

Текущий уровень задачи является одним из входных параметров, получаемых приемами; он учитывается решающими правилами приемов и позволяет организовать необходимые приоритеты в их применении: при меньших значениях этого уровня срабатывают приемы с большим приоритетом. В процессе обучения решающие правила приемов корректируются таким образом, чтобы на каждом шаге выбирался прием, наиболее целесообразный с точки зрения обучающего систему эксперта. При первоначальном обращении к задаче Z_i ее текущий уровень равен 0.

Изменение текущей ситуации происходит в следующем рабочем цикле решателя:

1) Происходит обращение к программе логического символа f — типа задачи Z_N. Если при этом не срабатывает ни один из приемов, либо сработавшие приемы изменили лишь комментарии задач и ни один из них не указал явно на необходимость повторного рассмотрения задачи (в таких случаях говорим, что процедура не внесла существенных изменений в текущую ситуацию), то переход к пункту 3, иначе — к пункту 2.

2) Если в результате срабатывания приема определился ответ на задачу Z_N либо был выдан отказ на нее, то возобновляется прерванный ранее прием решения задачи Z_{N-1}, в процессе реализации которого возникла задача Z_N. При $N = 2$ в этом случае выдается ответ либо отказ на решаемую задачу и возвращение в программу интерфейса решателя; при $N = 1$ - происходит выход из логической системы. При отсутствии ответа либо отказа на задачу Z_N текущий уровень этой задачи заменяется на 0, и переход к пункту 1. Это означает, что при наличии существенных изменений, внесенных приемом, решатель повторяет цикл анализа текущей ситуации с самого начала.

3) Осуществляется последовательный просмотр всех условий и посылок F задачи Z_N, вес v которых равен текущему уровню m_N этой задачи либо равен $m_N + 1$. Сначала просматриваются условия, затем посылки; порядок просмотра условий (посылок) - слева направо по соответствующим спискам задачи Z_N. Если $v = m_N$, то происходит однократный просмотр всех вхождений логических символов φ в F (слева направо); если же $v = m_N + 1$, то — серия таких просмотров, в процессе которых значение текущего уровня задачи Z_N полагается последовательно равным $0, 1, \ldots, m_N$. Для каждого рассматриваемого вхождения логического символа φ в F осуществляется обращение к программе логического символа φ (исходные данные этой программы содержат полную информацию о координатах вхождения символа φ в задачу Z_N). Если процедура не внесла существенных изменений в текущую ситуацию, то переход к рассмотрению очередного вхождения логического символа, иначе — к пункту 2. Если просмотр терма F закончился безрезультатно, то вес его увеличивается на 1 (новые либо видоизмененные посылки и условия задачи получают вес 0). Если просмотр всех условий и посылок задачи Z_N закончился безрезультатно, то текущий уровень задачи Z_N увеличивается на 1. Если в результате он становится больше, чем максимальный уровень M_N, то на задачу Z_N выдается отказ, иначе — переход к пункту 1.

Использование весов посылок и условий позволяет сузить область просмотра при поиске очередного приема, исключая из нее посылки и условия, имеющие большой вес (они уже были достаточно хорошо рассмотрены ранее, и срабатывание связанного с ними приема маловероятно). В результате происходит локализация рассмотрения задачи, направляемого

в первую очередь на новые либо видоизмененные посылки и условия; рассмотрение же всей задачи в целом имеет место, как правило, лишь на начальном этапе ее решения. По мере повышения текущего уровня m_N задачи Z_N в просмотр вовлекаются ранее отложенные посылки и условия F, вес v которых больше m_N; это происходит при $v = m_N+1$ (см. пункт 3), причем предварительно осуществляется поиск приема, срабатывающего при рассмотрении F для меньших, чем m_N, значений текущего уровня. Переключение внимания при рассмотрении задачи может происходить также в результате срабатывания приемов, уменьшающих веса тех или иных условий и посылок.

3. Запуск решения задачи и его пошаговый просмотр

Запуск решения задачи происходит из просмотра списка задач некоторого концевого раздела задачника. Войдя в этот список и создав в нем новую задачу либо выбрав для решения одну из ранее имевшихся задач, следует прежде всего добиться, чтобы верхняя горизонтальная линия данной задачи была прорисована на экране, а верхняя линия предыдущей задачи — не была видна на экране. Альтернативный способ — выделение нужной задачи нажатием правой кнопки мыши на ее поле (чтобы задача оказалась выделена, ее верхняя горизонтальная линия опять же должна быть видна на экране), после чего можно произвольно перемещаться по списку задач и выполнять запуск выделенной задачи из любой его точки.

Если требуется получить ответ задачи без отображения процесса решения, то нажимается клавиша “o” (кир.). Если на задачу будет получен ответ, то он будет прорисован в верхней части экрана, с указанием времени решения (в секундах либо минутах), а также с указанием трудоемкости, измеряемой в числе шагов работы интерпретатора языка ЛОС (этот язык нижнего уровня, используемый для записи приемов, описывается в последующих разделах книги). Ответ и трудоемкость сохраняются в файлах задачника и впоследствии будут прорисовываться непосредственно под условием задачи (время решения не сохраняется). Для исключения их из файлов следует выделить задачу (правой кнопкой мыши) и нажать “Ctr-F4”. Если ответ на задачу не получен, то происходит перерисовка ее текста с указанием под ним времени, затраченного на решение.

Если нужно отображать процесс решения по шагам, то нажимается клавиша “p” (кир.). Каждый последующий шаг обеспечивается нажатием “Enter”. Если в некоторый момент нужно оборвать пошаговое отображение решения и далее решать задачу до получения ответа, то нажимается

клавиша “0” (ноль). Если нужно вообще прервать решение, то нажимается клавиша “Esc”, возвращающая к тексту задачи в задачнике.

При показе очередного шага решения в верхней части экрана прорисовывается описание текущего действия. Над этим описанием отображается вся цепь задач (кроме фиктивной исходной задачи)- сначала идет выбранная из задачника задача (возможно, измененная в процессе решения по сравнению с ее первоначальной версией, оставшейся в задачнике), затем вспомогательная задача, к которой произошло обращение от первой задачи, и т.д. Под последней задачей этой цепи задач и размещается описание текущего действия. При просмотре всех этих записей применяется та же прокрутка, что и при просмотре списка задач в задачнике.

Отображаемые на экране задачи приводятся с теми же сокращениями, что и в задачнике (для включения либо выключения полного просмотра нажимается клавиша “ы”). Кроме того, в просматриваемой цепи задач могут встречаться “замаскированные” под задачи обращения к вспомогательным процедурам (пакетным операторам, см. описание языка ГЕНОЛОГ), не являющиеся задачами. Такие вспомогательные операторы введены из соображений оптимизации: каждый из них содержит в себе сравнительно небольшое число приемов, и поиск нужного приема в нем ускоряется на порядок по сравнению с поиском по всей базе приемов. Типы этих операторов аналогичны типам задач: проверочный оператор является аналогом задачи на доказательство, нормализатор — аналогом задачи на преобразование, синтезатор — аналогом задачи на описание и анализатор — аналогом задачи на исследование.

Под кадром, содержащим описание текущего действия, могут быть размещены несколько кадров со вспомогательными задачами — эти задачи решались в процессе выполнения данного действия. Можно повторно запустить решение любой из них — выделив ее либо разместив ее верхнюю линию в верхней части экрана и нажав “Enter”. Эту процедуру можно применять любое число раз и внутри пошагового просмотра решения вспомогательных задач — таким образом создается подобие гипертекста решения. Для возвращения на предыдущий уровень данного гипертекста нажимается клавиша “End”.

Комментарии к очередному действию решателя размещаются в скобках после описания этого действия. Вспомогательные задачи, сопровождающие текущее действие, также снабжаются комментарием, размещаемым в скобках перед описанием задачи. Иногда эти комментарии специально подготовлены для объяснения примененного приема, иногда они просто представляют собой подзаголовок того раздела оглавления базы приемов, в котором размещен примененный прием.

Если комментарий недостаточен для понимания выполненного действия или вообще непонятен (такое может случиться, если комментарий просто является подзаголовком концевого пункта оглавления приемов — некоторые из этих подзаголовков понятны только в контексте заголовков внешних разделов), то можно перейти к просмотру описания сработавшего приема. Конечно, здесь понадобится знакомство с языками, на которых задаются приемы решателя — ЛОСом и ГЕНОЛО́Гом (им посвящены последующие разделы книги). Для этого нажимается клавиша “б”, переводящая в просмотр приема (если прием реализован на языке ГЕНОЛОГ) либо (если прием реализован на языке ЛОС) переводящая в тот концевой пункт оглавления программ, который связан с последней пройденной перед реализацией приема контрольной точкой. В обоих случаях можно также просмотреть ЛОС-программу примененного приема, нажав клавишу “ф” — она переводит в отладчик ЛОСа. Для возвращения в просмотр текущего действия решателя из просмотра описания приема на ГЕНОЛОГе следует нажать клавишу “End”. Для возвращения из просмотра ЛОС-программы приема следует нажать клавишу “з”.

При просмотре текущего шага решения можно посмотреть исходную задачу в задачнике, не прерывая процесса решения. Для этого достаточно нажать клавишу “Home”; возвращение к просмотру текущего шага решения из задачника — по нажатии “End”.

Если при пошаговом просмотре возник промежуточный результат, представляющий самостоятельный интерес (даже в случае, когда ответ на решаемую задачу так и не будет получен), то можно выделить этот результат (целую формулу или ее часть, одну или несколько) — так же, как это делалось в задачнике, перейти в задачник (через “Home”) и зарегистрировать выделенную формулу в любой из задач или введя новый бланк задачи. Как и обычно, для этого следует войти в формульный редактор и нажать “Insert” - “N”, где N — номер выделенного элемента среди всех выделенных элементов; $1 \leq N \leq 9$. Далее можно нажать “End” и продолжить просмотр шагов решения.

Если возникла необходимость прервать процесс пошаговой трассировки решения некоторой задачи из цепи задач и перейти к очередному шагу для ее надзадачи, то следует выделить (правой кнопкой мыши) данную надзадачу и нажать “Enter”. Тогда следующий отображаемый на экране шаг будет относиться уже к выбранной надзадаче.

Если требуется прервать затянувшийся шаг решения задачи, то нажимается клавиша “Break”. В результате появится текст текущего выполняемого фрагмента ЛОС-программы, в котором текущий (еще не выполненный) оператор выделен малиновым цветом. Далее возможен анализ текущей ситуации с помощью средств отладчика ЛОСа (см. последующие разделы)

либо возвращение в главное меню при нажатии клавиши Esc. Можно также просмотреть текущую цепь задач (нажатием клавиши “з”) либо (до нажатия “з” либо вернувшись из просмотра цепи задач в ЛОС-программу нажатием “ф”) запустить процесс пошаговой трассировки на уровне той задачи, которая при прерывании оказалась текущей. Это делается нажатием клавиш “пробел” и “Enter”.

Иногда пошаговый просмотр решения оказывается неудобен из-за того, что на некотором этапе начинается решение трудоемкой вспомогательной задачи, обращение к которой не было вынесено в самостоятельный шаг. Разумеется, по завершении этого решения и отображении срабатывания приема, в рамках которого оно происходило, решение вспомогательной задачи можно повторить для ознакомления с подробностями. Однако, длительная пауза из-за неотображаемого процесса решения может оказаться нежелательной. В таких случаях можно повторно запустить пошаговый просмотр решения, изменив его режим непосредственно перед появлением паузы. Для выбора нужного режима трассировки следует нажать клавишу “т” (это делается либо до запуска решения, из задачника, либо уже в начатом процессе трассировки). После нажатия появляется диалог установки режима.

Для ввода либо отмены условия на трассировку, определяемого в одном из пунктов диалога, следует переместить курсор мыши внутрь прямоугольника этого пункта и нажать левую клавишу мыши (наличие плюса справа от прямоугольника означает, что условие включено, иначе — выключено). После выбора необходимой комбинации условий, курсор мыши перемещается в прямоугольник “Ввести” и снова нажимается левая клавиша мыши.

Для преодоления указанных выше пауз можно воспользоваться пунктом “ручной выбор входа в подпроцесс”. Если этот пункт активирован, то каждая попытка обращения решателя к вспомогательной задаче (включая наиболее крупные пакетные операторы) будет вводиться на экран. Чтобы продолжить решение без входа в пошаговую трассировку этой вспомогательной задачи, нажимается “Enter” ;чтобы войти в нее, нажимается “Ctr-Enter”. Заметим, что в этом режиме текущее действие решателя не сопровождается выдачей списка вспомогательных задач, решенных для его выполнения — сообщения об обращениях к этим задачам уже выдавались на экран до осуществления действия, и была возможность просмотреть их решение по шагам. Недостатком режима с ручным входом в подпроцессы является чрезмерное количество выдаваемых на экран сообщений о попытках решения вспомогательных задач, бóльшая часть которых оказывается неудачной. Действительно ценные

шаги тонут в этом потоке сообщений. Поэтому данный режим является лишь техническим, в обычных ситуациях отключенным.

Отладочные режимы трассировки описываются в разделе, посвященном отладчику ЛОСа. С помощью этих режимов можно, в частности, обеспечить прерывание процесса решения при попытке применения заданного приема, что часто используется при оптимизации его решающих правил.

Возможен серийный запуск решения задач из задачника. Такой запуск бывает необходим для контроля за изменениями в поведении решателя при его обучении. Простейшая форма этого запуска — выбор в оглавлении задачника нужного подраздела и нажатие клавиши “Ctr-з” либо клавиши “Ctr-э”. В первом случае из цикла решения будут исключены все задачи, которые при предыдущем запуске решателем не были решены; во втором случае будут решаться все задачи подряд.

При серийном решении последовательно решаются сначала все задачи из текущего пункта просматриваемого меню (выделенного в момент запуска) оглавления задачника, затем — все задачи из следующего пункта этого меню, и т.д. до конца подраздела. На экране при этом последовательно сменяются формулировки решаемых задач. Если решатель слишком долго решает какую-либо задачу (возможно, “залипает” на ней из-за плохих решающих правил), то последовательное нажатие клавиш “Break” и “Esc” переводит в решение следующей задачи. Если до конца решения всех задач подраздела произойдет “зависание” программы и понадобится ее перезапуск, то после перезапуска автоматически восстанавливается прерванный цикл решения задач — вплоть до достижения конца меню.

Для обрыва серийного режима следует нажать клавишу “Break” (на экране появится фрагмент ЛОС-программы, в котором произошло прерывание, и установится пошаговый режим отладочной трассировки), и далее — нажать клавишу “Ctr-з”. Лишь после этого нажатие клавиши “Esc” выведет в главное меню.

По окончании серийного запуска обновляется статистика о результатах решения задач подраздела. Такая статистика позволяет находить “особые точки” в задачнике (например, выявлять задачи, которые перестали решаться или решение которых замедлилось после внесенных в приемы изменений). Для просмотра статистики следует войти в меню нужного подраздела и нажать клавишу “з”. После небольшой паузы, необходимой для просмотра всех задач подраздела, на экране возникает таблица, в которой указываются следующие сведения:

а) Пять наибольших величин замедления в решении задач по сравнению с предыдущим запуском. Эти величины измеряются в числе шагов

интерпретатора ЛОСа, как и величины трудоемкости решения задач, сохраняемые в задачнике — отсюда легко извлекается процент замедления. Если нужно просмотреть имеющие самые большие значения замедления задачи, то нажимается клавиша “з”, переводящая в просмотр первой из этих задач. Переход к очередной задаче (отобранные для просмотра задачи упорядочены по убыванию замедлений) осуществляется нажатием клавиши “ш”. Для просмотра в таком режиме отбираются не более 40 задач подраздела (это же ограничение распространяется на другие приводимые ниже случаи отбора серий задач).

б) Число нерешенных задач раздела. Для просмотра этих задач нажимается клавиша “н”. Чтобы перейти к просмотру очередной нерешенной задачи, следует нажать “ш”.

в) Число утерянных решений задач подраздела (только по сравнению с предпоследним циклом решения). Для просмотра серии таких задач сначала нажимается “у”, и далее — через нажатия “ш”.

г) Пять наибольших величин ускорения в решении задач. Просмотр задач с ускорившимся решением — через нажатие клавиши “с”, и далее к каждой следующей задаче — через нажатие “ш”.

д) Суммарное замедление в решении задач подраздела (отрицательная его величина означает суммарное ускорение).

е) Число изменившихся по сравнению с предпоследним циклом решения ответов на задачи подраздела. Для просмотра серии соответствующих задач сначала нажимается клавиша “и”, и далее — через нажатия “ш”.

ж) Число сомнительных ответов. Для просмотра серии соответствующих задач — нажатие “о”, и далее — через “ш”.

з) Число задач, замедлившихся более чем на 10000 шагов интерпретатора ЛОСа, и число задач, замедлившихся более чем на 100000.

и) Можно просмотреть список всех ответов на задачи подраздела. Для этого нажимается клавиша “О”. Если на выделенном ответе нажать клавишу “курсор вправо”, то происходит переход к просмотру условия соответствующей задачи.

к) Пять наибольших величин трудоемкости решения задач подраздела (в числе шагов интерпретатора). Для просмотра задач в порядке убывания трудоемкости (не более 40 задач) — нажатие “т”, и далее — через “ш”.

Серийный запуск позволяет косвенно оценивать “холостой ход” приема — суммарную трудоемкость попыток его применения, не приводивших к срабатыванию приема. Фактически здесь оценивается относительная трудоемкость попыток применения приема на одно его срабатывание. Чтобы

получить такую статистику, требуется запустить серийное решение не через “Ctr-з”, а через “Ctr-х” (кир.). Тогда в указанной выше таблице статистики будут отображены данные о пяти наихудших случаях такой относительной трудоемкости (“холостого хода”). Эти данные суть пары (суммарная трудоемкость попыток применить прием — число срабатываний приема), для которых отношение первого элемента ко второму (если второй равен 0, то вместо него берется 1) наибольшее. Возможен просмотр серии наихудших приемов (по убыванию указанной характеристики) — через нажатие клавиши “х” (кир.). Переход к каждому очередному приему — через “ш”. При просмотре приема повторный вызов на экран указанной пары чисел — через “Х” (кир.); пара (A_1, A_2) прорисовывается в виде двух термов — “пассив(A_1)” и “актив(A_2)”.

4. Параллельная прокрутка решателя по задачнику

Прокрутку по задачнику можно распараллелить. Для этого следует заблаговременно создать в той же директории, где находится файл logsyst.exe, поддиректории EX1,EX2,...,EXn. Число n не превосходит уменьшенного на 1 числа потоков, реализуемых процессором машины. При этом оно не должно превосходить 23. В названии поддиректорий сначала n записывается как цифра от 1 до 9, а далее — как латинская буква начиная с a. “Наибольшая” допустимая буква — n.

В каждой поддиректории EXn полностью копируются все директории GEN, INF, LOS, TER, TXT, TCH. К ним добавляется файл logsyst.exe, в названии которого буква t заменена числом (или буквой) n.

Для корректной прокрутки следует установить на компьютере пакет AutoIt, который бесплатно скачивается в интернете. Он необходим для автоматического перезапуска программой fenix.exe тех потоков, в которых при прокрутке произойдет непредвиденный сбой. Кроме того, нужно позаботиться, чтобы Windows не заблокировала главному потоку доступ к боковым потокам (установить для всех потоков разрешение доступа “Все”).

Собственно запуск параллельной прокрутки по задачнику начинается с того, что в некотором (возможно, корневом) меню клавишей Ctr-1 выбирается начальный раздел прокрутки, а затем клавишей Ctr-2 — последний раздел. В случае корневого меню не рекомендуется выбирать последним раздел с номером, большим 17. Если используются 2 машины, нажимается Ctr-3, если 3 машины — Ctr-4. Иначе нажатие пропускается. Далее нажимается Ctr-л. На экране появятся небольшие окна для потоков, в

которых будет отображаться текущая трудоемкость задачи. Отобранные задачи распределяются по этим потокам равномерно, и каждая порция прокручивается независимо. По мере завершения прокрутки окна потоков исчезают. Если поток остановлен операционной системой, а программа fenix.exe его не перезапустила, такой перезапуск необходимо выполнить вручную. Главный поток, после завершения прокрутки, приобретает зеленый цвет. Пока все потоки не завершатся, возвращение в однопотоковый режим не происходит. Перед возвращением выдерживается пауза, необходимая для пересылки данных из боковых потоков в главный. Дальнейший анализ итогов прокрутки — такой же, как в последовательном режиме прокрутки.

5. Упражнения по вводу и решению задач

Ввести перечисленные ниже задачи в соответствующие разделы задачника и просмотреть процесс их решения.

5.1. Элементарная алгебра

1) Упростить выражение:

$$\sqrt{\frac{4}{x}+\frac{1}{4x^{-1}}-2}+\sqrt{\frac{1}{4x^{-1}}+\frac{2^{-2}}{x}+\frac{1}{2}}$$

2) Решить уравнение:

$$\frac{x^2}{x+2}+1=\frac{4}{x+2}$$

3) Решить неравенство:

$$\frac{\sqrt{x^2+x-6}+3x+13}{x+5}>1$$

4) Решить систему уравнений:

$$\begin{cases} (x+y)(x^2-y^2) &= 16 \\ (x-y)(x^2+y^2) &= 40. \end{cases}$$

5) Решить уравнение:

$$(tgx)^{\cos^2 x}=(ctgx)^{\sin x}$$

6) Решить неравенство:

$$|x-2|^{\log_4(x+2)-\log_2 x} < 1$$

7) Для всех a решить неравенство $ax^2 + (a+1)x + 1 > 0$.

8) При каких a неравенство $\sin^6 x + \cos^6 x + a \sin x \cos x \geq 0$ выполнено для всех значений x?

9) Найти все a, при которых из неравенства $x^2 - a(1+a^2)x + a^4 < 0$ следует неравенство $x^2 + 4x + 3 > 0$.

10) При каких a система

$$\begin{cases} x^2 + y^2 &= 2(1+a) \\ (x+y)^2 &= 14 \end{cases}$$

имеет ровно два решения?

5.2. Указания

1) Находясь в главном меню, выбрать пункт “Оглавление задачника” (клавиша “з” либо левая кнопка мыши в прямоугольнике указанного пункта). Используя клавиши курсора, найти в корневом меню оглавления раздел “Элементарная алгебра” (возможно, сначала понадобится несколько раз нажать “курсор влево”). Войти в этот раздел, далее войти в подраздел “Упрощение выражений”, затем — в любой из подразделов “Упрощение иррациональных выражений — 1,2,3”. В действительности выбор раздела несущественен; решатель будет работать вне зависимости от того, в каком разделе находится задача, и классификация задач по разделам нужна лишь для упрощения поиска нужной задачи вручную. Используя “PageDown” либо “курсор вниз”, вывести на экран последний пункт выбранного концевого раздела (все его пункты - номера с тремя штрихами). Затем нажать клавишу “к” (кир.) для создания бланка новой задачи. Далее нажимается “ц” и в оглавлении типов целевых установок выбираются разделы “Преобразовать выражение” — “Упростить выражение в области допустимых значений”, причем после выбора последнего пункта нажимается “курсор вправо”. Экран расчищается, и в верхней его части возникает текст “Упростить в о.д.з. выражение:”. Далее нажимается “Enter”, и формульным редактором вводится упрощаемое выражение. В процессе набора можно получать справки о клавиатуре формульного редактора, нажимая F1. Точнее, нажатие F1 переводит в общее оглавление

справочника по системе, и из его корневого меню нужно перейти в раздел “Формульный редактор”. Далее полезно прочитать раздел “общие сведения”; для получения информации о вводе конкретных символов — переходить в соответствующие подразделы. Чтобы вернуться в набор формулы, достаточно нажать “End”. По завершении набора упрощаемого выражения нажимается “Enter”. В данном примере задача оказывается полностью введенной, и для решения ее без пошаговой трассировки нажимается “о” (кир.), а для входа в пошаговую трассировку (она отображает лишь верхний уровень процесса решения — срабатывания приемов; такую трассировку, в отличие от отладочной трассировки, называем далее семантической) нажимается “р”. Каждый очередной шаг трассировки — нажатие “Enter”. Если в некоторый момент под кадром, поясняющим текущее действие, оказываются расположены другие кадры, то эти последние суть кадры обращений к вспомогательным задачам, решавшимся для выполнения текущего действия. Можно выбрать любой из них для детального просмотра решения вспомогательной задачи. При этом верхняя отделяющая линия кадра должна быть в точности верхней границей экрана; такое выравнивание обеспечивается клавишами “Ctr-курсор вверх либо вниз”. Вход в решение вспомогательной задачи — нажатие “Enter”. По завершении трассировки решения вспомогательной задачи нажатие “Enter” возвращает на предыдущий уровень (такое возвращение можно обеспечить в процессе трассировки нажатием “End”).

2) Действия аналогичны предыдущим, но выбирается раздел “Элементарная алгебра” — “Решение уравнений” — “Рациональные уравнения 1,2”. Нажимаются “к”, “ц”, и выбирается раздел оглавления целевых установок “Найти значения неизвестных” — “Получить полное явное описание значений неизвестных”. На экране появляется текст “Найти”, под которым размещен курсор формульного редактора. Этим редактором вводятся неизвестные задачи (отделенные запятой) — в данном примере единственная переменная x. После нажатия “Enter” ввод целевой установки завершается. Для ввода уравнения снова нажимается “Enter”, и далее происходит набор уравнения.

3) Действия аналогичны предыдущим, но выбирается подраздел для неравенств.

4) Для ввода системы уравнений и (или) неравенств сначала вводится целевая установка (аналогично предыдущему, но число неизвестных может быть более одной). Затем по отдельности вводятся условия —

уравнения и неравенства; ввод каждого нового условия начинается с “Enter” и завершается “Enter”. После ввода последнего условия задача готова к запуску решателя.

5) Аналогично задаче 2; тригонометрические операции вводятся последовательным набором двух (в особых случаях трех) первых латинских букв обозначения этих операций: синус — s,i; косинус — c,o; тангенс — t,g; котангенс — c,t, и т.д. Заметим, что степень тригонометрической операции набирается нестандартным образом: сначала набирается вся операция, затем — курсор вверх, затем — показатель степени и “Enter”. Тригонометрическая операция относится к наименьшему расположенному после нее осмысленному выражению (суммы и произведения под такой операцией следует заключать в скобки), и степень оказывается относящейся не к аргументу операции, а ко всей операции.

6) Аналогично задаче 3; вертикальные отрезки модуля вводятся с помощью клавиш “Ctr-м” (лат.), для ввода логарифма последовательно вводятся буквы l,o, после чего набирается основание логарифма. После набора основания нажимается “Enter”, и набирается выражение под логарифмом (суммы и произведения следует заключать в скобки).

7) Аналогично предыдущей задаче; параметр a не включается в число неизвестных задачи, и никак более не выделяется.

8) Неизвестной задачи служит переменная a. Условие ее набирается в виде
$$\forall_x(x - \text{число} \to \sin x^6 + \cos x^6 + a \sin x \cos x \geq 0).$$
Для ввода “x − число” сначала вводится x, затем нажимается /, затем “ч”.

9) Аналогично предыдущему; условие набирается в виде:
$$\forall_x(x^2 - a(1 + a^2)x + a^4 < 0 \to x^2 + 4x + 3 > 0.$$
Заметим, что понятие “следует” здесь трактуется таким образом, что если левая от стрелки часть ложная (в частности, неравенство слева вообще не имеет решений), то вся импликация истинна. Это приводит к тому, что концевые точки указанного в ответе промежутка (для них левое неравенство не имеет решений) отнесены к ответу.

10) Условие задачи состоит в том, что множество пар (x, y), удовлетворяющих уравнениям, имеет мощность 2. Оно записывается в виде

$$card(set_{xy}(x^2 + y^2 = 2(1 + a)\ \&\ (x + y)^2 = 14)) = 2.$$

5.3. Планиметрия

1) Основание равнобедренного треугольника равно a, угол при вершине равен b. Найдите биссектрису, проведенную к боковой стороне.

2) В трапеции $ABCD$ с основаниями AD и BC имеем $AD = 3$, $BC = 1$. Точка P лежит на стороне AB, а точка Q — на стороне CD, причем отрезок PQ параллелен основаниям и проходит через точку пересечения диагоналей трапеции. Найти длину отрезка PQ.

3) Известно, что $ABCD$ — ромб и радиусы окружностей, описанных около треугольников ABC и ABD соответственно, равны R и r. Найти площадь ромба $ABCD$.

4) В параллелограмме $ABCD$ биссектриса угла BAD пересекает сторону CD в точке M, причем $DM/MC = 2$. Известно, что угол CAM равен a. Найти угол BAD.

5) Окружность проходит через вершины A и C треугольника ABC, пересекает сторону AB в точке D и сторону BC в точке E. Известно, что $AD = 5, AC = 2\sqrt{7}, BE = 4, BD/CE = 3/2$. Найти угол CDB.

5.4. Указания

1) Вычислительные задачи по геометрии вводятся в следующем порядке: сначала набираются посылки задачи, перечисляющие известные свойства чертежа; затем вводится целевая установка задачи; затем указываются неизвестные величины, которые должны быть вычислены. Перед набором посылок можно ввести чертеж, однако чертеж может быть создан системой и автоматически (по окончании набора задачи нажимается “Ctr-ч”).

   При наборе посылок следует обязательно обозначить все точки, участвующие в задаче, буквами (обычно большими; можно использовать натуральные индексы). Ссылаться на плоские фигуры (треугольники, многоугольники, окружности, прямые, и т.д.) можно только через их “базисные” точки. В нашем случае обозначим вершины треугольника буквами A, B, C и введем первую посылку “$\triangle(ABC)$”. Для ввода посылки нажимается “Enter”, затем последовательно нажимаются клавиши “т”, “р” (обычно используются первые

две буквы вводимого понятия). Это приводит к прорисовке символа $\triangle$ с расположенной после него открывающей скобкой. Далее последовательно нажимаются A,B,C и закрывающая скобка, после чего — завершающее набор формулы "Enter". Никакие разделители между буквами не ставятся. Заметим, что если в обозначении треугольника либо другой фигуры используется буква с индексом, то после такой буквы, перед набором следующей, обязательно нужно нажать на клавишу "умножение" (звездочка).

После ввода первой посылки (она указывает только на то, что три точки A, B, C образуют вершины некоторого треугольника, то есть не лежат на одной прямой) нужно ввести посылку, определяющую, что треугольник равнобедренный. Выберем в качестве основания треугольника вершины A, C и введем посылку $l(AB) = l(BC)$. Для обозначения расстояния дважды нажимается клавиша "l", что приводит к появлению рукописной латинской буквы l с идущей после нее открывающей скобкой. Затем (как в обозначении треугольника) подряд вводятся буквы для точек, между которыми рассматривается расстояние, и ставится закрывающая скобка.

Далее вводятся: посылка $l(AC) = a$, обозначающая длину основания треугольника через a, и посылка $\angle(ABC) = b$ — для величины угла при основании. Чтобы получить обозначение угла, последовательно нажимаются клавиши "у", "г". Далее — как для обозначения треугольника. Как и обычно, вершина угла размещается в обозначении угла посередине.

Для ввода основания D биссектрисы треугольника ABC, проведенной из угла BAC, можно использовать посылку "Биссектреуг($BACD$)". Другой способ (более громоздкий, но не использующий специального обозначения "Биссектреуг") — пара посылок "$D \in$ прямая(BC)", "биссектриса($BACD$)". В первом случае нажимаем "И", "Т", и далее — как в случае обозначения треугольника, но для четырех букв. Во втором случае сначала вводим D, затем нажимаем пробел, b, e (лат.; появляется символ $\in$). Далее нажимаем "п", "р" (кир.; появляется слово "прямая" с открывающей скобкой), вводим буквы B, C и закрывающую скобку. Для ввода "биссектриса(...)" нажимаем клавиши "и", "т".

На этом ввод посылок завершен. Для ввода целевой установки нажимаем клавишу "ц", переводящую в оглавление типов целевых установок. Из корневого меню этого оглавления переходим к пункту "Найти значения неизвестных" — "Выразить значения неизвестных через заданные параметры". Заметим, что применявшийся в элементарной алгебре пункт "Получить полное явное описание зна-

чений неизвестных" в планиметрических задачах на вычисление НЕ следует использовать. Это объясняется принципиальным различием понятий "известная" и "неизвестная" в задачах из двух этих разделов. В элементарной алгебре все переменные из списка посылок по умолчанию считались известными и могли входить в ответ задачи. В геометрической задаче на вычисление посылки содержат обозначения точек $A, B, C, \ldots$, которые не должны появляться в ответе. Соответственно различаются и типы выбираемых целевых установок. После выбора указанного типа целевой установки нажимается "курсор вправо". Далее сначала вводится буква (или несколько отделенных запятыми букв) для неизвестной (неизвестных) и нажимается "Enter". Затем вводятся разделенные запятыми буквы для известных числовых параметров, через которые должны быть выражены неизвестные (если таких параметров вообще нет, сразу нажимается "Enter", иначе оно нажимается после ввода параметров). В нашем примере обозначим неизвестную длину биссектрисы через x; параметры суть a, b.

После ввода целевой установки вводим равенства, связывающие неизвестные (переменные) с теми выражениями, которые они обозначают и которые нужно вычислить. В ответ войдет сама неизвестная, а не обозначаемое ею выражение.

Как уже говорилось выше, после ввода задачи по планиметрии можно ввести чертеж — либо попробовать сделать это автоматически (нажатием "Ctr-ч"), либо ввести чертеж вручную (нажать "ч" и далее воспользоваться геометрическим редактором; чертеж появится перед списком посылок, в начале задачи). Можно также скорректировать вручную чертеж, созданный автоматически (вход в редактирование уже созданного чертежа — снова через "ч"; удаление чертежа — выделить его левой кнопкой мыши и нажать "Ctr-Del").

2) Напомним, что в решателе предусмотрены два варианта обозначения трапеции: "трапеция($ABCD$)" и "Трапеция($ABCD$)" — первый из них для случая трапеции с большим основанием AD и острыми углами при основании; второй — для случая, когда углы при основании не обязательно острые. В обоих случаях указанные записи предствляют собой даже не обозначения трапеции, а лишь утверждения о том, что точки A, B, C, D являются вершинами соответствующей трапеции. Сама трапеция (как и любой другой четырехугольник) обозначается посредством выражения "фигура($ABCD$)". В нашем примере годится любой из указанных способов записи. Например, введем посылку "трапеция($ABCD$)". Затем вводятся по-

сылки $l(AD) = 3, l(BC) = 1$. Принадлежность точки P стороне AB, а точки Q — стороне CD, записываются в виде $P \in$ отрезок(AB), $Q \in$ отрезок(CD). Параллельность отрезка PQ основаниям трапеции записывается как прямая$(PQ)\|$прямая(AD). Чтобы сформулировать условие о том, что отрезок PQ проходит через точку пересечения диагоналей трапеции, нужно ввести обозначение для этой точки. Например, обозначим ее через M. То, что M является точкой пересечения диагоналей, записываем как $M \in$ прямая(AC), $M \in$ прямая(BD). Далее добавляем посылку $M \in$ отрезок(PQ). На этом ввод посылок завершается. Как и в предыдущей задаче, выбираем целевую установку и вводим единственное условие $x = l(PQ)$ для неизвестной x.

3) Начинаем со ввода посылки "ромб$(ABCD)$". Так как в задаче речь идет об описанных окружностях, надо ввести обозначения для этих окружностей, то есть обозначить центр окружности и выбрать какую-либо точку на окружности (в планиметрии ссылки на окружности могут быть только такими). Например, обозначим центры через M, N. В случае описанной окружности, которая должна проходить через вершины треугольника, в качестве второй точки можно взять какую-либо вершину треугольника (например, A). Тогда добавятся посылки "окружность(MA) описана около фигура(ABC)"; "окружность(NA) описана около фигура(ABD)". Заметим, что хотя эти тексты кажутся достаточно длинными, набираются они весьма малым числом нажатий клавиш: сначала нажимаем "о", "к" (кир.) — появляется слово "окружность" с открывающей скобкой. Затем вводим буквы M, A и закрывающую скобку. Далее нажимаем "Ctr-ф" — появляются слова "описана около". Наконец, нажимаем "ф", "и" — появляется слово "фигура" с открывающей скобкой. В заключение вводим A,B, C и закрывающую скобку. Радиусы окружностей указываем с помощью посылок $l(MA) = R, l(NA) = r$. При вводе целевой установки указываем, что значение неизвестной x должно быть выражено через R, r. Наконец, набираем условие $x = S($фигура$(ABCD))$. Заметим, что знак площади S здесь вводится двукратным нажатием малой латинской буквы s; если его ввести нажатием клавиши большой латинской S, то задача окажется набранной неверно и не будет решена (вместо площади окажется введенным значение какой-то неопределенной функции S).

4) Вводятся посылки "параллелограмм$(ABCD)$", "биссектриса$(BADM)$", $M \in$ отрезок(CD), $l(DM)/l(MC) = 2$, $\angle(CAM) = a$. Затем выбирается целевая установка "Выразить x через a", и вводится условие $x = \angle(BAD)$.

5) Вводятся посылки $\triangle(ABC)$, $C \in$ окружность(PA), $D \in$ окружность(PA), $D \in$ отрезок(AB), $E \in$ окружность(PA), $E \in$ отрезок(BC), $l(AD) = 5, l(AC) = 2\sqrt{7}, l(BE) = 4, l(BD)/l(CE) = 3/2$. Затем вводятся целевая установка и условие $x = \angle(CDB)$.

5.5. Математический анализ

1) Вычислить производную функции

$$\frac{e^{-x^2}\arcsin(e^{-x^2})}{\sqrt{1-e^{-2x^2}}} + \frac{1}{2}\ln(1-e^{-2x^2})$$

2) Вычислить предел функции

$$(2e^{\frac{x}{x+1}} - 1)^{\frac{x^2+1}{x}}$$

при $x \to 0$.

3) Исследовать поведение функции $y = (x-5)\sqrt[3]{x^2}$.

4) Найти точки экстремума для функции $y = x^2 - ln(x^2)$.

5) Исследовать на непрерывность функцию $y = \text{arctg}(1/x + 1/x - 1 + 1/x - 2)$.

6) Найти неопределенный интеграл

$$\int \frac{2\sin x - \cos x + 3}{3\sin x + \cos x + 1} dx$$

7) Вычислить определенный интеграл

$$\int_0^{ln2} \sqrt{e^x - 1} dx$$

8) Вычислить двойной интеграл от $\sqrt{|x-y^2|}$ по области $(x,y) : |y| \leq 1, 0 \leq x \leq 2$.

9) Найти площадь области, заключенной между параболой $y^2 = \frac{b^2}{a}x$ и прямой $y = \frac{b}{a}x$; $0 < a, 0 < b$.

10) Найти объем тела $x^2 \leq ay \leq bx, x^2 + y^2 \leq hz \leq 2x^2 + 2y^2$; $0 < a, 0 < b, 0 < h$.

11) Найти площадь поверхности $z = xy, x^2 + y^2 \leq R^2$.

12) При каких значениях параметра p сходится ряд

$$\sum_{i=1}^{\infty} \frac{1}{i^p} \sin \frac{\pi}{i}?$$

13) Разложить в ряд Тейлора по переменной x в точке 0 функцию

$$y = (x - \operatorname{tg} x) \cos x$$

14) Разложить в ряд Фурье на отрезке $[-\pi, \pi]$ функцию $y = (\pi^2 - x^2)^2$.

15) Найти сумму ряда

$$\sum_{n=1}^{\infty} \frac{3n^2 + 3n + 1}{n^3(n+1)^3}.$$

5.6. Указания

1) Прежде всего нужно войти в оглавление целевых установок и выбрать пункт "Упростить выражение в области допустимых значений". Затем набирается производная указанного выражения. Она вводится как дробь вида $\frac{dA}{dx}$; A - дифференцируемое выражение. После символа d нужно ставить знак умножения; дифференцируемое выражение заключается в скобки. В принципе, переменную d можно использовать и внутри выражения A, и даже взять ее в качестве x. Если нужно найти значение производной в некоторой точке t, то вместо dx набирается $d(x = t)$, причем и здесь после d ставится знак умножения.

2) Снова вводится установка "Упростить выражение в области допустимых значений". Затем набирается условие: сначала нажимаются клавиши l, i — появляется символ lim, справа внизу от которого размещен курсор. Набираем $x \to 0$ (стрелка вводится клавишей "курсор вправо") и нажимаем Enter — курсор перемещается вверх в ту же строку, в которой расположен символ lim. Здесь набираем выражение под знаком предела. Это выражение обязательно нужно заключить в скобки, иначе знак предела будет отнесен к минимальному осмысленному его началу. В нашем примере, если не заключить все выражение в скобки, то степень окажется вне предела и ответ будет другим.

3) Выбирается целевая установка "Исследовать поведение функции", в которой указывается обозначение функции — переменная y. Затем вводится условие $y = \lambda_x((x-5)\sqrt[3]{x^2}, x - \text{число})$.

4) Выбираем переменные, которые будут обозначать, соответственно, точку экстремума, значение функции в этой точке, и тип экстремума (максимум либо минимум). Например, пусть это будут переменные u, v, w. Вводим целевую установку "Найти полное явное описание значений неизвестных" в которой указываем выбранные неизвестные. Затем вводим условие

$$Extr(\lambda_x(x^2 - \ln(x^2), x - \text{число}), u, v, w).$$

Аналогично вводятся задачи на поиск множества точек минимума либо максимума некоторой функции внутри заданной области, но число неизвестных здесь будет равно двум (искомое множество точек и значение функции в этих точках). Вместо $Extr$ используются символы Min, Max, причем после функции должна быть указана область, по которой берется минимум или максимум.

5) Отличие от общего исследования поведения функции — только в том, что выбирается целевая установка "Исследовать функцию на непрерывность".

6) Выбирается целевая установка "Упростить выражение в области допустимых значений". Затем набирается неопределенный интеграл: нажимается "Ctr-j" и вводится подынтегральное выражение, которое умножается справа на произведение dx.

7) Аналогично неопределенному интегралу, но нажимается "Ctr-i". Тогда курсор сначала оказывается под знаком интеграла, где набирается нижний предел. После нажатия "Enter" курсор переводится вверх, где набирается верхний предел. Еще одно нажатие "Enter" — и набирается подынтегральное выражение, умножаемое на dx.

8) При вычислении двойных интегралов область интегрирования обычно задается в списке посылок. Выбираем для нее обозначение - например, переменную P, и введем посылку $P = set_{xy}(|y| \leq 1 \ \& \ 0 \leq x \ \& \ x \leq 2)$. Затем вводим целевую установку "Упростить выражение в области допустимых значений". Затем набираем двойной интеграл — нажимаем "Ctr-2"; под интегралом вводим обозначение области P, нажимаем "Enter", и далее набираем подынтегральное выражение, умноженное на произведение $dxdy$.

9) Плоскую область определяем в списке посылок, обозначив ее вспомогательной переменной (например, P). Эту область задаем, используя ссылку на прямоугольную систему координат, которую тоже обозначаем вспомогательной переменной (например, K). После этого в

нашем примере вводим посылки

$$P = \text{точки}(\text{областьграницы}(set_{xy}(y^2 = \frac{b^2}{a}x) \cup set_{xy}(y = \frac{b}{a}x)), K),$$

$0 < a$, $0 < b$, "прямкоорд(K)". Заметим, что операция "областьграницы" применяется к теоретико-множественному объединению пар координат точек кривых, ограничивающих область, а значением этой операции служит множество пар координат точек области. Для перехода от пар координат к точкам плоскости используется операция "точки". Далее выбирается целевая установка "Упростить выражение в области допустимых значений" и вводится условие $S(P)$. Как и в случае геометрических задач, символ площади S вводится двойным нажатием клавиши s.

10) В случае нахождения объемов применяется аналогичная конструкция, но множество троек координат трехмерной области задается непосредственно, с помощью неравенств. В нашем примере вводим посылки $P = \text{точки}(set_{xyz}(x^2 \le ay \;\&\; ay \le bx \;\&\; x^2 + y^2 \le hz \;\&\; hz \le 2x^2 + 2y^2), K)$, $0 < a$, $0 < b$, $0 < h$, "прямкоорд(K)". Целевая установка — та же, что и в предыдущем примере; условие имеет вид "объем(P)" (двукратное нажатие клавиши "О", кир.).

11) Посылки вводятся аналогично предыдущей задаче, причем вместо множества троек координат точек трехмерной области задается множество троек координат точек поверхности. Условие имеет вид $S(P)$.

12) Целевая установка при наличии параметров ряда — "Получить полное явное описание значений неизвестных" (при их отсутствии - "Проверить истинность утверждения"); неизвестная — параметр ряда p. Условие имеет вид утверждения о сходимости последовательности частичных сумм ряда:

$$\text{сходится}(\lambda_n(\sum_{i=1}^{n}(\frac{1}{i^p}\sin\frac{\pi}{i}), n - \text{натуральное}))$$

13) Выбирается установка "Разложить в ряд Тейлора", в которой указываются переменная разложения и точка разложения. Условием задачи служит выражение $(x - \operatorname{tg} x)\cos x$.

14) Аналогично предыдущему — с установкой "Разложить в ряд Фурье".

15) Установка — "Упростить выражение в области допустимых значений". Условие задачи набирается непосредственно в виде бесконечной суммы.

5.7. Аналитическая геометрия и линейная алгебра

1) Даны три вектора $a(4,1,5)$, $b(0,5,2)$ и $c(-6,2,3)$. Найти вектор x, удовлетворяющий системе уравнений $(x,a)=18, (x,b)=1, (x,c)=1$. Система координат прямоугольная.

2) Даны координаты двух вершин треугольника $A(-1,3), B(2,5)$ и точки пересечения его высот $H(1,4)$ в прямоугольной системе координат. Найти координаты третьей вершины треугольника и составить уравнения его сторон.

3) Составить уравнения плоскостей, проходящих через прямую $\frac{x-1}{3} = \frac{y-1}{5} = \frac{z+2}{4}$ и равноудаленных от точек $A(1,2,5)$ и $B(3,0,-1)$.

4) Составить уравнение касательной к параболе $y^2 = -8x$, отрезок которой между точкой касания и директрисой делится осью Oy пополам. Система координат прямоугольная.

5) Найти уравнение плоскости, пересекающей эллипсоид $x^2 + 2y^2 + 4z^2 = 9$ по эллипсу, центр которого находится в точке $C(3,2,1)$. Система координат прямоугольная.

6) Поверхность задана уравнением $6xy - 8y^2 - z^2 + 60y + 2z + 89 = 0$ в прямоугольной системе координат. Найти каноническую систему координат и каноническое уравнение этой поверхности. Определить тип поверхности.

7) Решить матричное уравнение:

$$\begin{pmatrix} 2 & -3 & 1 \\ 4 & -5 & 2 \\ 5 & -7 & 3 \end{pmatrix} \cdot X \cdot \begin{pmatrix} 9 & 7 & 6 \\ 1 & 1 & 2 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & -2 \\ 18 & 12 & 9 \\ 23 & 15 & 11 \end{pmatrix}$$

8) Вычислить определитель

$$\begin{vmatrix} \sin a & \cos a & \sin(a+d) \\ \sin b & \cos b & \sin(b+d) \\ \sin c & \cos c & \sin(c+d) \end{vmatrix}$$

9) Найти собственные значения и собственные векторы линейного преобразования, заданного матрицей:

$$\begin{pmatrix} 3 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 3 & 0 & 5 & -3 \\ 4 & -1 & 3 & -1 \end{pmatrix}$$

10) Найти каноническую жорданову форму для матрицы:

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

5.8. Указания

1) В задачах по аналитической геометрии обычно приходится вводить обозначение для используемой системы координат. Обозначим ее в нашем примере через K и введем посылку "прямкоорд(K)". Напомним, что в решателе системы координат на плоскости отождествляются с тройками точек общего положения (первая точка — начало системы координат, две последние — концы координатных векторов), а в пространстве — с четверками таких точек. Если в задаче имеются ссылки на координатные оси либо плоскости, то приходится явным образом вводить такие тройки либо четверки точек. В нашем примере этих ссылок нет, поэтому далее вводим посылки, определяющие координаты векторов a, b, c относительно K: "коорд$(a, K) = (4, 1, 5)$"; "коорд$(b, K) = (0, 5, 2)$"; "коорд$(c, K) = (-6, 2, 3)$". Затем вводим посылки "скалумнож$(a, x) = 18$", "скалумнож$(b, x) = 1$", "скалумнож$(c, x) = 1$". Выбираем целевую установку "Выразить значения неизвестных через заданные параметры" и вводим переменную y для неизвестной. Наконец, вводим условие задачи "$y =$ коорд(x, K)".

2) Вводим обозначение K для прямоугольной системы координат и заносим посылки "прямкоорд(K)", $\triangle(ABC)$, "коорд$(A, K) = (-1, 3)$", "коорд$(B, K) = (2, 5)$". Чтобы определить H как точку пересечения высот, проводим две высоты - из вершины A и из вершины B. Это можно сделать, добавив две посылки "Высота($ABCM$)" и "Высота($BCAN$)"; M, N — основания высот. Далее добавляем посылки, связанные с точкой H: $H \in$ прямая(AM); $H \in$ прямая(BN); "коорд$(H, K) = (1, 4)$". Выбираем в качестве неизвестных переменные x, y, z, v — для уравнений сторон треугольника и координат его третьей вершины. Наконец, создаем целевую установку (так же, как в предыдущей задаче) и вводим условия: $x =$ коорд(прямая$(AB), K)$, $y =$ коорд(прямая$(AC), K)$, $z =$ коорд(прямая$(BC), K)$,$v =$ коорд(C, K).

3) В этой задаче система координат не предполагается прямоугольной. Поэтому выбираем обозначающую ее переменную K, но никаких

специальных посылок для K не вводим. Так как в условии задачи говорится о прямой, а прямые в решателе обозначаются только с помощью пары точек, выбираем переменные P, Q для обозначения этих точек. После этого можно ввести посылку, задающую уравнение прямой: "коорд(прямая$(PQ), K) = set_{xyz}$(пропорцнаборы$((x - 1, y - 1, z + 2), (3, 5, 4)))$". Заметим, что использование отношения "пропорцнаборы" пропорциональности двух числовых наборов при задании канонического уравнения прямой в пространстве является для решателя стандартом. Чтобы обозначить плоскость, выбираем переменные C, D, E для трех точек общего положения на этой плоскости. Затем вводим посылку, выражающую включение прямой PQ в искомую плоскость: "прямая$(PQ) \subseteq$ плоскость(CDE)". Вводим посылки, определяющие координаты указанных в задаче точек A, B: "коорд$(A, K) = (1, 2, 5)$", "коорд$(B, K) = (3, 0, -1)$". Наконец, указываем на равноудаленность плоскости CDE от точек A, B: "расстдоплоскости$(A,$ плоскость$(CDE)) =$ расстдоплоскости$(B,$ плоскость$(CDE))$". Затем вводим целевую установку (так же, как в предыдущих задачах), указывая в ней неизвестную x для координат плоскости CDE. Завершаем набор задачи условием $x =$ коорд(плоскость$(CDE), K)$.

4) Так как система координат прямоугольная, вводим посылку "прямкоорд(K)". Обозначаем параболу через E и указываем ее уравнение: "коорд$(E, K) = set_{xy}(y^2 = -8x)$". Выбираем обозначение "прямая(AB)" для касательной и вводим посылку "прямая$(AB) -$ касательная к E". Для точки касания можно было бы выбрать специальное обозначение, однако без ограничения общности можно считать, что этой точкой служит точка A. Поэтому вводим посылку $A \in E$. Для директрисы вводим обозначение "прямая(FG)". Так как в решателе упоминание о директрисе связано с обязательным упоминанием фокуса, которому она соответствует, то выбираем обзначение C для фокуса и заносим посылки "фокус(C, E)", "директриса(прямая(FG),$C, E)$". В качестве точки пересечения касательной с директрисой можно без ограничения общности выбрать точку B; соответствующая посылка имеет вид $B \in$ прямая(FG). Для обозначения середины отрезка AB выбираем переменную D и заносим посылки $D \in$ отрезок(AB), $l(AD) = l(BD)$. Чтобы указать на принадлежность точки D оси ординат, вводим явное обозначение для тройки точек, образующих систему координат $K : K = (M, N, P)$. Теперь принадлежность точки D оси ординат записывается в виде $D \in$ прямая(MP). Чтобы предотвратить недопустимый в задаче случай совпадения прямой AB с осью ординат,

добавляем посылку $\neg(A \in \text{прямая}(MP))$. Далее вводим целевую установку (как в предыдущей задаче) с упоминанием неизвестной x и добавляем условие "$x = \text{коорд}(\text{прямая}(AB), K)$".

5) Вводим посылки "прямкоорд(K)" и "коорд$(E, K) = set_{xyz}(x^2 + 2y^2 + 4z^2 = 9)$";
E — обозначение эллипсоида из условия задачи. Обозначаем искомую плоскость "плоскость(ABC)", а сечение ею эллипсоида — F, после чего вводим посылки $E \cap \text{плоскость}(ABC) = F$ и "эллипс(F)". Обозначив центр эллипса через D, добавляем посылки "центр(D, F)" и "коорд$(D, K) = (3, 2, 1)$". Далее вводим целевую установку с неизвестной x и условие "$x = \text{коорд}(\text{плоскость}(ABC), K)$".

6) Вводим посылку "прямкоорд(K)" и выбираем целевую установку "Исследовать свойства поверхности, заданной своим уравнением". В этой установке обозначаем исследуемую поверхность через E, и вводим условие задачи: "коорд(E, K)
$= set_{xyz}(6xy - 8y^2 - z^2 + 60y + 2z + 89 = 0)$".

7) Для решения матричного уравнения используем обычную целевую установку уравнений "Получить полное явное описание значений неизвестных". Для набора матрицы нажимаем клавиши "м", "а" (кир.) — появляется левая скобка матрицы. Затем вводится первая строка, причем после каждого элемента нажимается "Enter". Чтобы перейти к следующей строке, после набора последнего элемента вместо "Enter" нажимается "Page Down". После набор последней строки нажимается "End" — возникает правая скобка матрицы. Для отката в случае ошибочных действий при наборе матрицы используется "Backspace". После набора первой матрицы вводится обычный символ умножения (клавиша "звездочка"), затем X, затем снова умножение, далее — вторая матрица, и после знака равенства — третья. Преобразование умножения чисел в матричное умножение предпринимается автоматически при запуске решения задачи.

8) Для вычисления определителя выбирается целевая установка "Упростить выражение в области допустимых значений". При наборе условия сначала нажимаются клавиши "d", "e" (лат.) - возникает символ det. Затем вводится матрица.

9) Выбираем обозначение для матрицы — переменную A. Затем вводим посылку — равенство с переменной A в левой части и матрицей в правой. Выбираем в качестве неизвестных переменные x, y, z; x — собственное значение, y — его кратность, z - собственный вектор, отвечающий данному собственному значению. Затем выбираем

целевую установку “Получить полное явное описание значений неизвестных” для указанных неизвестных и вводим условия “собствзначение(A, x, y)”, “собствектор(A, x, z)”. В ответе для z приводится условие принадлежности множеству линейных комбинации базиса собственных векторов, соответствующих собственному значению x.

10) Выбираем целевую установку “Получить полное явное описание значений неизвестных” для неизвестных x (жорданова форма) и y (матрица, преобразующая к жордановой форме, т.е. такая, что произведение обратной к ней матрицы, исходной и снова матрицы y, равно x). Затем вводим условие “жордформа(A,x,
y)”, где A — исходная матрица.

5.9. Дифференциальные уравнения

1) Решить уравнение

$$2(y - 2xy - x^2\sqrt{y}) + x^2y' = 0$$

2) Решить уравнение

$$xyy'' + xy'^2 - yy' = 0$$

3) Найти решения уравнения $y = xy' - 2y'^2$, проходящие через точку (4,2).

4) Найти решение системы дифференциальных уравнений:

$$\begin{cases} \frac{dx}{dt} &= x - y + 4\cos(2t) \\ \frac{dy}{dt} &= 3x - 2y + 8\cos(2t) + 5\sin(2t). \end{cases}$$

5.10. Указания

1) Выбираем целевую установку “Решить функциональные уравнения” (эта установка берется и для остальных задач на решение дифференциальных уравнений). В качестве неизвестной указываем уже не переменную y, а выражение $y(x)$. Затем вводим условие задачи — уравнение, в котором везде вместо функции y записывается ее значение $y(x)$ в точке x, а вместо производной y' - $\frac{dy(x)}{dx}$. Напомним, что по всем произвольным постоянным, возникающим при интегрировании уравнения, в ответе навешивается квантор существования.

2) Задача вводится аналогично предыдущей; производная второго порядка набирается в виде $\frac{d^2y(x)}{dx^2}$. Символ дифференцирования d здесь рассматривается как переменная — то есть после него или его степени нужно нажимать клавишу “звездочка” для умножения.

3) После ввода дифференциального уравнения добавляем условие $y(4) = 2$.

4) В целевой установке указываем две неизвестных функции - $x(t), y(t)$. Уравнения системы набираются аналогично предыдущим задачам; вместо x, y в них используются записи $x(t), y(t)$.

6. Логический калькулятор

Для ускоренного ввода и решения задач создан так называемый логический калькулятор. Вход в него происходит через пункт “Решить задачу” главного меню системы. После выбора нужного концевого пункта возникающего здесь оглавления нужно зайти в него нажатием “курсор вправо” и далее действовать согласно появляющимся инструкциям. В нижней части экрана размещаются сведения по набору формул рассматриваемого раздела. Если этих сведений недостаточно, следует нажать F1, перейти в справочник по системе и из его корневого меню использовать раздел “Формульный редактор”. Иногда может быть полезен также раздел “Логический язык системы”.

По завершении ввода задачи решатель сохраняет ее в разделе “Буфер — Последние задачи” оглавления задачника и сразу приступает к решению. После получения ответа либо отказа можно продолжить работать с сохраненной задачей стандартными средствами - запустить просмотр шагов решения, перенести задачу в другой раздел и т.п. Для расчистки буфера достаточно, находясь в любой точке задачника, нажать “Shift-o” (“o” кириллица).

Разумеется, возможности логического калькулятора ограничены текущими средствами решателя. Иногда они достаточно для решения задач стандартных типов, иногда — нет. Последние случаи даже более ценны, так как подсказывают возможности дальнейшего пополнения базы приемов.

7. Анализ траекторий решения задач при обучении решателя

Пополнение базы приемов решателя происходит при ручном обучении только за счет анализа траекторий решения задач. Создавать какие-либо приемы из общих соображений, без примерки на задачах, не рекомендуется. Велика вероятность того, что такой прием не будет использоваться

решателем — либо из-за того, что он окажется избыточным и его срабатывание предвосхитят другие приемы, либо из-за того, что действия других приемов уведут задачу из области его применимости. Обучающий материал для решателей дают задачники, а не учебники.

Таким образом, чтобы научиться создавать решатели, прежде всего нужно овладеть техникой разбиения траектории решения задачи на последовательность применений приемов. Отсюда возникают исходные неформальные версии описаний приемов, которые затем уже записываются на ГЕНОЛОГе или на ЛОСе. Разумеется, для одной и той же задачи могут быть предложены сильно отличающиеся друг от друга способы решения, а для одного и того же решения — различные версии объясняющих его приемов. Наиболее простые и эффективные варианты не всегда удается "угадать" по единственному примеру, так что оптимизация решателя предполагает более или менее регулярные откаты для модернизации уже созданных групп приемов. Обычно эти откаты имеют локальный характер, а использование компилятора ГЕНОЛОГа делает их в техническом отношении не слишком дорогостоящими.

Рассмотрим несколько примеров анализа траекторий решения задач, в которых не будем предполагать наличия каких-либо ранее созданных приемов. Все эти задачи взяты из задачника решателя, и в качестве упражнения можно рекомендовать проследить по шагам его действия. Приводимые ниже рассуждения дают лишь первое, весьма приблизительное представление о разбиении решений на приемы. Реальное обучение решателя требует учета гораздо большего числа технических подробностей.

Начнем с простейшего примера — решения уравнения

$$\frac{6a+7b}{6a} - \frac{3bx}{2a^2} = 1 - \frac{bx}{a^2-ab}.$$

Первый шаг, который представляется естественным в этой задаче - группировка в левой части всех членов с неизвестной x, а в правой — всех известных членов. Этот шаг сразу подсказывает прием: если обе части числового уравнения имеют неизвестные либо известные слагаемые, то выполняется указанная выше перегруппировка членов. Уровень срабатывания приема можно взять совсем маленьким, например, равным 1. Для усиления стандартизации вводим еще один прием, переводящий неизвестную часть равенства влево, а известную — вправо. Уровень его пусть тоже будет равен 1. Каких-либо соображений об ограничении применения данных приемов не возникает. Однако, следует учитывать, что теперь на уровнях выше первого все уравнения будут иметь известные слагаемые

только в правой части. Поэтому, например, шаблон для усмотрения квадратных уравнений должен иметь вид $ax^2 + bx = c$, вместо привычного $ax^2 + bx + c = 0$.

Итак, получаем уравнение:

$$\frac{bx}{a^2 - ab} - \frac{3bx}{2a^2} = 1 - \frac{6a + 7b}{6a}.$$

Следующий шаг — сложить дробные слагаемые в левой части уравнения. Формулируем соответствующий прием: "если левая часть уравнения имеет дробное слагаемое, то обращаемся к вспомогательной задаче на преобразование ее к виду дроби, после чего выполняем замену". Каких-либо оснований откладывать это действие не видно, так что уровень срабатывания приема снова выбираем равным 1. Целесообразность применения данного приема уже не столь очевидна, как в предыдущем случае. Сложение дробных выражений может привести к очень громоздкому результату, и задача будет заведена в тупик. С другой стороны, сразу привести условия, отделяющие допустимые применения от недопустимых, мы не можем. Чтобы возникли какие-то подсказки на этот счет, нужно все-таки ввести прием таким, как есть, и подождать появления задачи, в которой он будет мешать. Тогда и начнется накопление серии дополнительных эвристических ограничений, которые обеспечат должное управление приемом.

В нашем случае эта работа уже проделана — можно заглянуть в решатель и посмотреть, какие ограничения возникли. Прежде всего, оказалось, что для систем уравнений уровень срабатывания данного приема лучше положить равным не 1, а 3. Если нужно не складывать неизвестные дробные выражения, а обозначить их новыми неизвестными и решить полученную вспомогательную систему, то прием, выполняющий эти действия, успеет сработать. Если уравнение содержит неизвестную, явно выраженную с помощью еще одного уравнения через другие неизвестные, то прием блокируется — лучше (вообще говоря) сначала подставить найденное значение. Блокировка происходит также при наличии в уравнении неизвестного логарифма от суммы с дробным слагаемым. Она заставляет решатель сначала преобразовать указанный логарифм. Наконец, в случае единственной неизвестной применение приема откладывается (как и в случае систем) до уровня 3 при наличии неизвестных логарифмов по разным основаниям. Перечисленные условия относятся к сравнительно редким ситуациям, и таким образом применение рассматриваемого приема оказалось "почти всегда" оправданным.

Введенный нами прием обратился к вспомогательной задаче на сложение дробных выражений. Поэтому временно прерываем анализ цепочки

преобразований основной задачи и переходим к рассмотрению выражения

$$\frac{bx}{a^2 - ab} - \frac{3bx}{2a^2}.$$

План наших действий очевиден — сначала нужно разложить на множители знаменатели, а затем выполнить сложение дробей. Оформим эти действия в виде приемов.

Разложение на множители числителей и знаменателей можно считать преобразованием общей стандартизации, предшествующим применению других приемов, относящихся к дробям. Однако, на этапе завершающего редактирования упрощаемого выражения может понадобиться обратное преобразование — иногда раскрывание скобок приводит к получению более компактной записи. Момент перехода к завершающему редактированию ответа можно помечать вводом специального комментария задачи, и тогда первый прием (разложение на множители) будет срабатывать при отсутствии данного комментария, а второй (попытка упрощающего раскрывания скобок) — при его наличии. Точкой применения приема обращения к разложению на множители можно считать сумму - основание степени, являющейся сомножителем знаменателя (соответственно, числителя) дроби, допуская вырожденный случай степени с показателем единица. В нашем примере единственной такой точкой является выражение $a^2 - ab$.

Прием, применяемый здесь для разложения на множители, состоит в вынесении за скобку общего множителя всех слагаемых. Для его программирования понадобится вспомогательная процедура, находящая наибольший общий делитель двух одночленов.

После разложения на множители знаменателя получаем выражение

$$\frac{bx}{a(a - b)} - \frac{3bx}{2a^2}.$$

Прием, выполняющий сложение дробных выражений, сначала находит общий множитель числителей и общий множитель знаменателей. Они сразу выносятся за скобки, после чего применяется обычное преобразование: числители и знаменатели перемножаются “крест накрест” и складываются. Перед тем, как составить результирующую дробь, предпринимается попытка разложить эту сумму на множители. Последнее действие может показаться излишним — ведь уже имеется прием, который пытается разложить на множители числитель. Однако, тогда понадобятся два цикла сканирования задачи вместо одного — система как бы “забудет” о том, что только что сложила дроби, и лишь после нового цикла рассмотрения задачи натолкнется на указанный числитель. Циклы сканирования

задачи в решателе обычно составляют главную часть вычислительного времени. Поэтому лучше объединять в одном и том же приеме как основное действие, так и все сопровождающие его дополнительные — это дает весьма ощутимое ускорение.

В нашем случае задача решается с целью сложить дроби, и каких-либо особых эвристических решающих правил не требуется. Достаточно ввести в прием проверку наличия данной цели. Впрочем, можно ввести ограничение, требующее при сложении нескольких дробных выражений начинать с самых коротких. Иногда это упрощает выкладки.

Возвращаемся к цепочке преобразований уравнения, которое после сложения дробных выражений в левой части приобретает вид

$$\frac{bx(3b-a)}{2(a-b)a^2} = 1 - \frac{6a+7b}{6a}.$$

Это уравнение имеет в левой части дробь; для устранения ее естественно домножить обе части уравнения на знаменатель. Однако, если сформулировать прием подобным образом, то он иногда будет приводить к излишним вычислительным затратам. Целесообразно до исключения знаменателя левой части сложить дробные выражения в правой, известной части. Тогда можно будет сначала вынести за скобку общие множители числителей и знаменателей, и лишь затем избавляться от знаменателей. Результатом преобразований станет дизъюнкция, объединяющая в себе уравнение с исключенными знаменателями и частные случаи равенства нулю общих множителей числителей:

$$b = 0 \vee 3x(3b-a) = -7a(a-b).$$

Как и прием сложения дробных выражений, данный прием исключения знаменателей объединяет сразу несколько независимых преобразований, ускоряя тем самым процесс вычислений.

Следующий шаг решения задачи — разбор случаев. Прием, используемый для этого, последовательно рассматривает уравнения, соответствующие подслучаям, и затем объединяет полученные ответы связкой “или”. Ответ каждого подслучая упрощается отдельно, однако дизъюнкция ответов, вообще говоря, допускает дальнейшее упрощение — какие-то серии корней или целые промежутки могут склеиваться. Поэтому прием должен обращаться к вспомогательной задаче на упрощение полученной дизъюнкции. Более того, этот же прием должен сразу выдать ответ задачи, иначе дизъюнкция, которая им получена, снова вызовет разбор случаев, и система зациклится.

В нашем примере первый подслучай — условие $b = 0$. Оно не содержит неизвестных, и может быть сразу выдано в качестве ответа. Это действие, хотя и очень простое, тоже требует специального приема. Кроме усмотрения того, что условия задачи не содержат неизвестных, данный прием должен обратиться к вспомогательной задаче на упрощение их конъюнкции, и результат упрощения выдать в качестве ответа.

Второй подслучай — уравнение $3x(3b-a) = -7a(a-b)$. Согласно условиям на область допустимых значений (хотя мы и опустили их рассмотрение, но приведенные выше приемы должны были сопровождать все преобразования коррекцией таких условий), правая его часть отлична от нуля. Поэтому уравнение эквивалентно соотношению

$$x = -\frac{7a(a-b)}{3(3b-a)}.$$

Формулировка приема, выполняющего данный переход, очевидна.

Далее следует подстановка найденного значения неизвестной в сопровождающие условия на область допустимых значений и упрощение этих условий. Эти действия требуют специального приема, выполняющего обращение к задаче на редактирование ответа. Полученный ответ

$$a \neq 0, a-b \neq 0, 3b-a \neq 0, x = \frac{7a(b-a)}{3(3b-a)}$$

возвращается приему разбора случаев, объединяющему его с ответом $a \neq 0, b = 0$ первого подслучая и выдающему окончательный ответ.

Разобранный пример оказался совсем простым с точки зрения управления преобразованиями — каждое действие было практически однозначным. Рассмотрим несколько более сложный случай — решение системы уравнений. Будем решать систему:

$$\begin{cases} x^2 + y^2 &= z^2 \\ xy + yz + xz &= 47 \\ (z-x)(z-y) &= 2. \end{cases}$$

Поначалу каких-то соображений однозначного характера о ее преобразованиях не возникает. Однако, есть соображения о том, с чего начинать анализ ситуации. Например, можно попробовать раскрыть скобки в третьем уравнении и посмотреть, не станут ли очевидными следующие шаги. Это — тоже преобразование, однако не основной задачи на описание, а сопровождающей ее задачи на исследование, в которой накапливаются общие следствия посылок и условий. Здесь мы имеем сразу два приема: первый принимает решение о переходе к выводу следствий в задаче на

исследование, второй - выводит из третьего уравнения следствие, получаемое раскрыванием скобок. Впрочем, в действительности эти приемы в решателе переставлены местами — раскрывание скобок выполняется сразу же. Случаи, когда оно может повредить, отслеживаются приемами, срабатывающими на меньших уровнях. Создавая прием для раскрывания неизвестных скобок в уравнениях, мы не должны сразу же предусматривать какие-то ограничения на его применение. Если они необходимы, то проявятся позднее, при анализе других задач. Так или иначе, получаем в списке посылок задачи на исследование первые два уравнения, сопровождаемые уравнением $xy - xz - yz + z^2 = 2$. Теперь становится видно, что при сложении второго и третьего уравнений уничтожаются сразу два неизвестных слагаемых в левой части. Получается уравнение $2xy+z^2 = 49$ — следствие двух исходных уравнений. Прием, выполняющий это действие, можно несколько обобщить — чтобы он искал линейную комбинацию двух уравнений, позволяющую устранить сразу два неизвестных слагаемых. Следующий естественный шаг — сложить первое уравнение с полученным следствием, чтобы исключить неизвестную z. Формулировка соответствующего приема несложна. После сложения уравнений имеем следствие $x^2+y^2+2xy = 49$. Очередной шаг очевиден - представить левую часть как полный квадрат: $(x + y)^2 = 49$. Прием, выполняющий это действие, обращается к вспомогательной задаче на разложение левой части уравнения на множители. Если такое разложение удается, то есть вероятность, что полученное уравнение будет полезно для дальнейшего (например, чтобы поделить два уравнения, сократив неизвестный множитель). Разумеется, нужно принять меры, чтобы прием не пытался разложить на множители левую часть уравнения, полученного после раскрывания скобок, и обратно. Для этого можно использовать специальные комментарии к уравнениям, блокирующие “обратный ход”. Следующий шаг — решение простейшего степенного уравнения. После него появляется дизъюнкция $x+y = 7 \vee x+y = -7$. Обычно получение дизъюнкции в задаче на исследование означает, что следует вернуться в исходную задачу на описание и предпринять разбор случаев. В нашей ситуации нужно, кроме того, учесть, что найденная дизъюнкция эквивалентна, при наличии первых двух уравнений, третьему. Это позволяет при разборе случаев исключить третье уравнение. Таким образом, цикл вывода следствий из уравнений завершился, и далее решаем две независимых системы. Ограничимся рассмотрением первой из них -

$$\begin{cases} x + y &= -7 \\ x^2 + y^2 &= z^2 \\ xy + yz + xz &= 47. \end{cases}$$

Очередной шаг состоит в преобразовании первого уравнения к виду $x = -(y + 7)$. Прием, реализующий это шаг, можно было бы представить как обращение к вспомогательной задаче, разрешающей одно из уравнений системы для исключения неизвестной. На первый взгляд, такое обобщение нашего шага может показаться естественным. Однако, при рассмотрении последующих примеров выяснилось бы, что данный прием почти всегда вреден — если произвольно выбрать какое-то уравнение системы и разрешить (пусть даже успешно) относительно одной из неизвестных, то чаще всего возникает такое усложнение системы, после которого решить ее становится весьма затруднительно. Сравнивая случаи, где выражение одной неизвестной через другие полезно, со случаями, где оно вредно, можно было бы создать несколько приемов, применяемых в очень специальных ситуациях. Для нашей задачи вводим прием, выражающий неизвестную из уравнения, если оно линейно по всем своим неизвестным.

Далее подставляем найденное выражение для x через y и переходим к решению системы относительно y, z. Это делается отдельным приемом, который должен создать вспомогательную задачу для двух неизвестных и обратиться к ее решению. Завершающая обработка ответа допускает разные варианты. Можно получить ответ на вспомогательную задачу, объединить его с уравнением $x = -(y+7)$ и упростить результат. Однако, если при решении вспомогательной задачи происходил разбор случаев и ответ на нее имеет вид дизъюнкции, то при упрощении снова понадобится разбор случаев. Поэтому в решателе реализован другой способ - уравнение $x = -(y + 7)$ передается вспомогательной задаче через ее технические структуры данных и извлекается оттуда при редактировании ответа для каждого подслучая. Тогда после решения вспомогательной задачи какой-либо дополнительной обработки ответа не потребуется.

После перехода к неизвестным y, z выполняются простые преобразования, связанные с общей стандартизацией вида уравнений и с раскрыванием скобок. Они реализуются простыми приемами, на которых можно сейчас не останавливаться. В результате возникает система $14y + 2y^2 - z^2 = -49, 7y+7z+y^2 = -47$. Легко заметить, что члены с неизвестной y в обоих уравнениях пропорциональны, и после вычитания из первого уравнения удвоенного второго остается уравнение с единственной неизвестной z. Прием, усматривающий возможность получить уравнение с единственной неизвестной за счет линейной комбинации уравнений, практически не требует каких-либо дополнительных ограничений на целесообразность срабатывания. В результате получаем систему $14y+2y^2-z^2 = -49, -14z-z^2 = 45$. Дальнейшие действия очевидны, и мы их разбирать не будем.

Разобранные примеры продемонстрировали два режима работы решателя — эквивалентные преобразования уравнения в первом случае и вывод следствий для получения дополнительной информации о неизвестных во втором. Первый режим, априори, требует большой осмотрительности при выборе каждого действия, так как неудачное преобразование может завести задачу в тупик. В действительности, однако, для многих областей наблюдается явление устойчивости: если в целом следовать некоторым несложным представлениям о том, какие преобразования упрощают задачу, то выбор конкретного порядка их выполнения несущественен — в любом случае за разумное число шагов получается один и тот же ответ. Это и позволяет в таких областях "избавляться от перебора". Второй режим позволяет исключать перебор в его классическом виде рассмотрения дерева задач, сводя поиск решения к рассмотрению расширяющегося списка посылок одной и той же задачи. Так как приемы, выводящие следствия, снабжены решающими правилами, отсекающими малоперспективные (например, чрезмерно громоздкие) с точки зрения эксперта следствия, то через определенное время наступает полное исчерпание разумных следствий, и процесс обрывается. Процедура решения задачи, вместо перебора, приобретает здесь характер "логического замыкания" исходных данных.

Рассмотрим пример, в котором возникает еще одна разновидность "логического режима". Будем решать задачу на доказательство неравенства

$$0 \leq a^4 - 2a^3b + 2a^2b^2 - 2ab^3 + b^4.$$

В таких задачах часто помогает прием, использующий неравенство для среднего арифметического и среднего геометрического. Удобнее переформулировать его в виде приема, выделяющего в оцениваемой сумме квадрат суммы или квадрат разности. В нашей задаче можно заметить, что слагаемые a^2b^2, b^4 и $-2ab^3$ представляют собой квадрат разности величин ab, b^2. Поэтому, если доказать неравенство $0 \leq a^2b^2 + a^4 - 2a^3b$, полученное отбрасыванием данных слагаемых, то задача будет решена. Применяя к последнему неравенству тот же прием, получаем искомое доказательство.

Заметим, что выбор группы слагаемых, образующих квадрат суммы или разности, выполняется неоднозначно. Например, на первом шаге можно было бы выделить группу $2a^2b^2, a^4, b^4$ и сразу завести задачу в тупик. По этой причине может показаться, что прием выделения оцениваемой группы слагаемых требует какого-то сильного управления, без чего возникнет трудоемкий перебор. Однако, данный прием обычно сильно упрощает правую часть неравенства, так что длина цепочек неравенств, возникающих при доказательстве, невелика. Кроме того, часто появляются

тупиковые ситуации, в которых прием неприменим. Поэтому реальный перебор оказывается невелик, и особых проблем с принятием решения не возникает. Здесь возникает режим "ограниченного перебора", у которого ограничения обусловлены не какими-то специальными решающими правилами, а просто тем фактом, что сложность задачи при каждом шаге сильно уменьшается. Этот режим используется в решателе для разных разделов (например, вычисление пределов и интегрирование), и обычно дает очень быстрое получение ответа.

Перейдем к примерам из других разделов. Рассмотрим сначала простую геометрическую задачу на вычисление. В параллелограмме $ABCD$ из точки N пересечения диагоналей проведены перпендикуляры NF, NE к сторонам AB,AD. Длины этих перпендикуляров равны, соответственно, p, m. Угол BAD равен a. Найти длины x, y диагоналей AC, BD и площадь параллелограмма z.

Схема решения геометрических задач на вычисление напоминает схему решения систем уравнений — происходит вывод следствий из посылок и условий до тех пор, пока не возникают либо равенства для значений неизвестных, либо уравнения, из которых значения неизвестных извлекаются чисто алгебраическими методами. В начале процесса вывода порождаются совсем простые утверждения, которые можно рассматривать как логическое представление чертежа задачи. В нашем примере таковыми являются утверждения о параллельности прямых AB, CD и AD, BC. При решении геометрической задачи последовательность рассуждений допускает множество вариаций. Итоговая картина при этом обычно оказывается не очень чувствительной к ее конкретной версии. Приведем для нашего примера одну из таких возможных последовательностей.

Прежде всего, замечаем, что в задаче нужно найти площадь параллелограмма и что к стороне AD проведен перпендикуляр NE. Для получения высоты, длина которой участвует в формуле площади, продолжаем перпендикуляр до пересечения с противоположной стороной в точке G, и одновременно выписываем соотношение $z = l(AD)l(EG)$. Эти действия оформляем в виде отдельного приема. Основаниями для его применения служат указанные выше признаки - упоминание в задаче о площади параллелограмма и "недоведенный" до конца перпендикуляр к одной из сторон.

Продолжая общий анализ чертежа, выводим из равенства $\angle(BAD) = a$ соотношения для четырех остальных углов параллелограмма. Основанием для применения этого приема является упоминание в задаче хотя бы одного из углов параллелограмма.

Аналогичным образом, выводим равенство длин противоположных сторон параллелограмма $l(BC)$ и $l(AD)$. Основанием применения приема служит упоминание в задаче одной из этих длин.

Так как выделена точка N пересечения диагоналей параллелограмма, замечаем, что диагонали делятся в ней пополам: $l(CN) = l(AN)$, $l(DN) = l(BN)$. Еще один прием отмечает, что точка E не просто лежит на прямой BD, но является точкой отрезка BD. Основанием для его срабатывания служит наличие в посылках задачи равенства $l(DN) = l(BN)$. Этот же прием устанавливает, что точка N лежит на отрезке AC.

Так как точка N находится на отрезке AC, причем длина отрезка и расстояния от N до его концов упоминаются в задаче, то выводим соотношение равенства длины отрезка сумме длин подотрезков: $x = 2l(AN)$. Аналогично, выводим $2l(BN) = y$.

Далее выводится условие принадлежности точки N отрезку EG — проведенной высоте параллелограмма. Основаниями являются принадлежность точки N диагонали и параллельность сторон. Отсюда, аналогично предыдущему, получается следствие $l(EG) = 2m$ - прием о равенстве длины отрезка сумме длин подотрезков срабатывает уже в третий раз.

Так как угол BAD и высота $l(EG)$ теперь известны, можно выписать тригонометрическое соотношение $\sin(a)l(AB) = 2m$, в котором единственный неизвестный числовой параметр - длина отрезка AB. Выписывание таких соотношений (возможно, не на самых малых уровнях сканирования задачи) представляется разумным, так как постепенно доопределяются новые параметры чертежа.

После того, как предыдущее соотношение занесено в посылки задачи, оно преобразуется к виду $l(AB) = 2m/\sin(a)$. Здесь работает прием, разрешающий линейное соотношение относительно числового параметра, определенного через ссылки на точки.

Как только оказалась введена в рассмотрение длина стороны AB, срабатывает уже упоминавшийся выше прием, выписывающий равенство длин сторон AB,CD.

Далее вводится высота параллелограмма CH, проведенная к продолжению стороны AB. Основанием для этого действия служит то, что, во-первых, длины отрезков AN,NC пропорциональны, а длина p высоты FN — известна. Отсюда прием выводит следствие $l(CH) = 2p$. Во-вторых, основанием для срабатывания является наличие известного угла CBA, который мог бы позволить выразить впоследствии через $l(CH)$ какие-то новые параметры чертежа.

Условие перпендикулярности прямых CH и AB преобразуется в условие параллельности прямых CH и FN - чтобы в дальнейшем иметь дело с классами параллельных прямых, выбирая в каждом таком классе единственного представителя.

Снова применяется прием, выписывающий тригонометрическое соотношение

$$\sin(a)l(AD) = 2p.$$

Однако, это другой прием, срабатывающий на меньшем уровне, чем рассмотренный выше. Оснований для его срабатывания больше — параметр $l(AD)$ к этому моменту уже встречается в задаче. Введенное соотношение сразу преобразуется к виду $l(AD) = 2p/\sin(a)$.

Теперь начинает срабатывать прием, подставляющий найденное значение $l(AD)$ во все посылки, где этот параметр встречается. В частности, таким образом из посылки $z = 2ml(AD)$ получаем фрагмент ответа $z = 4mp/\sin(a)$. Накопление информации о параметрах чертежа "само собой" привело к нахождению искомой площади. Разумеется, существенную роль при этом сыграли приоритеты на получение соотношений, устанавливающих связь данных и искомых параметров. Вывод следствий имеет характер "встречного распространения" цепочек зависимостей — от известных и от неизвестных величин.

К текущему моменту определились длины сторон параллелограмма и его углы. Поэтому для нахождения неизвестных диагоналей и завершения решения остается воспользоваться теоремой косинусов. Основания для срабатывания приема — известны длины двух сторон треугольника и угол между ними, а длина третьей стороны связана с неизвестными задачи каким-либо соотношением.

Перечисленные выше действия в действительности совпадают с действиями решателя. Таким образом, видно, что даже после накопления большой базы приемов удается избежать чрезмерного количества выводимых следствий. Это достигается тенденцией вводить при обучении как можно более сильные ограничения на срабатывания, пропускающие лишь действительно разумные шаги.

Сразу заметим, что при обучении решателя геометрическим задачам было допущено одно отклонение от "человеческих" стандартов. Вместо того, чтобы в цикле предварительного анализа чертежа определять, какие его элементы можно выразить через другие, и лишь затем выписывать нужные соотношения, решатель вводит эти соотношения сразу. Однако, за исключением редких случаев, он никак их не преобразует, а использует лишь для установления самого факта взаимной выразимости параметров - как своего рода "граф" взаимосвязей. На быстродействии компьютерной

системы это сказывается мало, но у пользователя, анализирующего ход решения по шагам, складывается впечатление об избыточной громоздкости накапливаемой информации. Впрочем, за счет дальнейшего усиления решающих правил, с данным явлением можно достаточно эффективно бороться. Иногда оно и вовсе незаметно.

Следующая задача — на качественное исследование поведения функции $f(x) = x^x$ с помощью пределов и производных. Она решается по схеме, аналогичной только что разобранному геометрическому примеру. Решение заключается в последовательном выводе следствий, характеризующих функцию f, и в отборе тех из них, которые целесообразно включить в итоговый список. Прежде всего применяется прием, находящий область определения функции. Он выписывает условия на область допустимых значений x и обращается к вспомогательной задаче на упрощение класса таких значений. Выводится следствие $\mathrm{Dom}(f) = (0, \infty)$. Следующий шаг — вычисление производной. Получаются следствия $g(x) = (1 + \ln x)x^x$; "Производная(f, g)". Снова применяется прием, определяющий область определения введенной в рассмотрение функции g. После того, как явно указаны области определения f, g, вводится посылка, указывающая множество точек, где производная не определена либо не вычислена: $\mathrm{Dom}(f) \setminus \mathrm{Dom}(g) = \emptyset$. Следующий шаг - срабатывание приема, обращающегося к вспомогательной задаче для отыскания корней производной и вводящего следствие $\mathrm{roots}(g, \mathrm{Dom}(g)) = \{1/e\}$. Чтобы анализировать поведение функции в точках, где ее производная определена и отлична от нуля, эта область явно находится и разбивается на промежутки. Соответствующий прием обращается к вспомогательной задаче, осуществляющей такое разбиение, после чего появляется посылка "областьроста$(f, (0, 1/e) \cup (1/e, \infty))$", распадающаяся на утверждения "областьроста$(f, (0, 1/e))$", "областьроста$(f, (1/e, \infty))$". Используя информацию о промежутках монотонности и анализируя знак производной на их стыке (точка $1/e$), следующий прием выводит утверждения "возрастает$(f, (1/e, \infty))$", "убывает$(f, (0, 1/e))$". Далее применяется прием, анализирующий знаки функции f на интервале монотонности. Он выводит следствия о числе корней на интервале: $\mathrm{card}(\mathrm{roots}(f, (0, 1/e))) = 0$, $\mathrm{card}(\mathrm{roots}(f, (1/e, \infty))) = 0$, немедленно преобразуемые к виду $\mathrm{roots}(f, (0, 1/e)) = \emptyset$, $\mathrm{roots}(f, (1/e, \infty)) = \emptyset$. Следующий прием усматривает экстремум на стыке промежутков монотонности -

$$\mathrm{Extr}(f, \frac{1}{e}, \frac{1}{\exp \frac{1}{e}}, \min).$$

На этом поток вывода следствий исчерпывается, и выдается ответ, в который отбираются: соотношение, определяющее функцию f, информация

об экстремуме, о промежутках монотонности, об области определения функции f и о числе ее корней на промежутках монотонности.

В этом примере управляющая компонента приемов была почти вырожденной — по существу, совокупность приемов составляла алгоритм исследования функции. Однако, разбиение такого алгоритма на отдельные почти не связанные друг с другом фрагменты — приемы - предоставляет несомненные удобства для последующего пополнения его все новыми и новыми элементами, ориентированными на различные специальные случаи.

Еще один пример, решаемый по схеме вывода следствий — из аналитической геометрии. Пусть стороны треугольника заданы уравнениями $7x + y - 2 = 0, 5x + 5y - 4 = 0, 2x - 2y + 5 = 0$. Нужно найти координаты точки внутри треугольника, равноудаленной от первых двух прямых и отстоящей от третьей на расстояние $\frac{3\sqrt{2}}{4}$. В аналитической геометрии, а в особенности в таких разделах, как теория вероятностей и физика, обычной математической символики для полной формулировки задачи оказывается уже недостаточно. Поэтому пошаговому анализу решения здесь предшествует анализ возможных вариантов логической формализации условия и отбор наиболее удобных вариантов. В нашем примере обозначения возьмем из логического языка решателя. Пусть вершины треугольника обозначены A, B, C, а прямоугольная система координат, относительно которой берутся уравнения — K. Сами уравнения можно записать тогда, например, в следующем виде: “коорд(прямая(AB),K) = $\text{set}_{xy}(x - \text{число}\ \&\ y - \text{число}\ \&\ 7x + y - 2 = 0)$”; “коорд(прямая($AC$),$K$) = $\text{set}_{xy}(x - \text{число}\ \&\ y - \text{число}\ \&\ 5x + 5 - 4 = 0)$”; “коорд(прямая($BC$),$K$) = $\text{set}_{xy}(x - \text{число}\ \&\ y - \text{число}\ \&\ 2x - 2y + 5 = 0)$”. Искомую точку обозначим D; условие ее принадлежности треугольнику запишем в виде $D \in$ фигура(ABC). Равноудаленность точки от первых двух прямых представляем записью “расстдопрямой(D,прямая(AB))= расстдопрямой(D,прямая(AC))”. Указываем расстояние до третьей прямой: “расстдопрямой(D,прямая(BC)) = $\frac{3\sqrt{2}}{4}$”. Перечисленные утверждения образуют список посылок задачи, т.е. то, что дано. Условием ее служит равенство $z =$ коорд(D, K), причем переменная z является неизвестной задачи. По постановке задачи, в выражение для z не должны входить обозначения A, B, C, D, K, хотя они и появляются в посылках задачи, а следовательно, формально являются “известными”. Это специально оговаривается в целевой установке задачи.

Процесс решения задачи выглядит как вывод следствий из объединенного списка посылок и условий. Приведем цепочку выводов, реализуемую решателем. Хотя она, быть может, и не оптимальна, однако для новичка вполне допустима.

Прежде всего, вводятся координаты точек A, B, C: “коорд(A,K)” $= (a, b)$; “коорд(B,
K)” $= (c, d)$; “коорд(C,K)” $= (e, f)$. Основанием для такого действия служит то, что точки встречаются в обозначениях прямых, для которых известны уравнения. Следующий шаг — ввод координатного набора для неизвестной z: $z = (g, h)$. Основанием является то, что в задаче рассматривается расстояние от точки D до прямой, уравнение которой известно. Для всех введенных новых параметров $a, b, \ldots, h$ регистрируются посылки, указывающие, что параметры — числовые. Подставляя координаты точки A в уравнение прямой AB, получаем соотношение $b+7a-2 = 0$. Его преобразуем к виду $b = 2 - 7a$ и подставляем b во все остальные посылки задачи. Аналогичным образом, выводим соотношение $d+7c-2 = 0$ и преобразуем к виду $d = 2 - 7c$. Подставляя координаты точки A в уравнение прямой AC, находим $a = 1/5$. Все эти действия выполняются простыми приемами, срабатывающими без ограничений. Подставляя значение a в уравнение для b, находим $b = 3/5$. Для точек B, C выполняем аналогичные действия; в итоге получаем $c = -1/16$, $d = 2 + 7/16$, $e = -17/20$, $f = 33/20$.

Определив координаты точек A, B, C (быть может, они для дальнейшего и не понадобятся), решатель переходит к выводу соотношений для искомых координат g, h. Так как уравнение прямой AC известно, можно воспользоваться формулой для расстояния от точки до прямой и получить соотношение

$$50(\text{расстдопрямой}(D, \text{прямая}(AC)))^2 = 50gh+25g^2+25h^2-40g-40h+16.$$

Основанием для этого действия служит то, что указанное расстояние уже упоминается в задаче, а уравнение прямой известно. Выражаем квадрат расстояния:

$$(\text{расстдопрямой}(D, \text{прямая}(AC)))^2 = \frac{50gh + 25g^2 + 25h^2 - 40g - 40h + 16}{50}.$$

Аналогичным образом, для расстояния от D до прямой AB получаем:

$$(\text{расстдопрямой}(D, \text{прямая}(AB)))^2 = \frac{14gh + 49g^2 + h^2 - 28g - 4h + 4}{50}.$$

С учетом посылки задачи, указывающей равенство данных расстояний, выводим соотношение

$$\frac{50gh + 25g^2 + 25h^2 - 40g - 40h + 16}{50} = \frac{14gh + 49g^2 + h^2 - 28g - 4h + 4}{50}.$$

Далее переходим к расстоянию от D до прямой BC:

$$(\text{расстдопрямой}(D, \text{прямая}(BC)))^2 = \frac{-8gh + 4g^2 + 4h^2 + 20g - 20h + 25}{8}.$$

В это равенство подставляем значение расстояния, данное в формулировке задачи:

$$\frac{9}{8} = \frac{-8gh + 4g^2 + 4h^2 + 20g - 20h + 25}{8}.$$

Начинается цепочка общей стандартизации последних уравнений. После устранения знаменателей и сокращения они преобразуются к виду: $5g - 2gh - 5h + g^2 + h^2 = -4$, $g + 3h + 2g^2 - 3gh - 2h^2 = 1$.

Теперь начинается учет условия принадлежности точки D треугольнику. Здесь-то и используются найденные ранее координаты вершин. Условие расположения точки D по ту же сторону от прямой AB, что и точка C, дает неравенство $h + 7g - 2 \leq 0$. В случае прямой AC получаем $0 \leq 5g + 5h - 4$. В случае прямой BC - $0 \leq 2g - 2h + 5$.

Далее решатель усматривает систему из двух приведенных выше уравнений относительно числовых параметров g, h и решает ее вне общего контекста, учитывая также неравенства для g, h. В результате получается $g = 0, h = 1$, $z = (0, 1)$, и задача решена.

Процесс вывода следствий похож на решение задачи по элементарной геометрии, однако управление приемами здесь существенно проще.

Список литературы

[1] Подколзин А. С., “Введение в логические процессы. Представление задач в решателе”, *Интеллектуальные системы. Теория и приложения*, **29**:2 (2025), 5–138.

[2] Подколзин А. С., *Компьютерное моделирование логических процессов. Том 1. Архитектура и языки решателя задач.*, Физматлит, Москва, 2008, 1024 с.

[3] Подколзин А. С., *Компьютерное моделирование логических процессов. Том 2. Опыт обучения компьютерного решателя задач: логические приемы, алгебра множеств, комбинаторика и элементарная алгебра.*, ВИНИТИ РАН, Москва, 2015, 1153 с.

[4] Подколзин А. С., *Компьютерное моделирование логических процессов. Том 3. Опыт обучения компьютерного решателя задач: математический анализ, дифференциальные уравнения и элементарная геометрия.*, ВИНИТИ РАН, Москва, 2015, 1320 с.

[5] Подколзин А. С., *Компьютерное моделирование логических процессов. Том 4. Опыт обучения компьютерного решателя задач: аналитическая геометрия, линейная алгебра, теория вероятностей, комплексный анализ и другие разделы.*, ВИНИТИ РАН, Москва, 2017, 969 с.

[6] Подколзин А. С., *Компьютерное моделирование логических процессов. Том 5. Опыт обучения компьютерного решателя задач: элементарные физика и химия, шахматы.*, ВИНИТИ РАН, Москва, 2019, 938 с.

[7] Подколзин А. С., *Компьютерное моделирование логических процессов. Том 6. Опыт обучения компьютерного решателя задач: понимание естественного языка и анализ рисунков.*, ВИНИТИ РАН, Москва, 2019, 757 с.

[8] Подколзин А. С., *Компьютерное моделирование логических процессов. Том 7. Автоматическое создание приемов логической системы: классификация приемов решателя, логический ассемблер, компилятор спецификаций, создание тестовых приемов и доводка приемов.*, ВИНИТИ РАН, Москва, 2021, 739 с.

[9] Подколзин А. С., *Компьютерное моделирование логических процессов. Том 8. Автоматическое создание приемов логической системы: база теорем, характеризация теорем, создание спецификаций приемов.*, ВИНИТИ РАН, Москва, 2021, 515 с.

[10] Подколзин А. С., *Компьютерное моделирование логических процессов. Том 9. Автоматическое создание приемов логической системы: логический вывод в базе теорем.*, ВИНИТИ РАН, Москва, 2022, 1494 с.

Introduction to Logical Processes. General diagram of the Solver's functioning

Podkolzin A.S.

This article describes the general layout of the mathematical problem solver. It explains how the verification process works, how to run problem solvers, and how to explore step-by-step previews. A large number of suggestions for entering and solving problems are provided.

Keywords: mathematical problem solver, logical processes, logical language, logical formalization of problems.

References

[1] Podkolzin A. S., "Introduction to Logical Processes. Representation of Problems in the Solver", *Intelligent Systems. Theory and Applications*, **29**:2 (2025), 5–138 (In Russian)

[2] Podkolzin A. S., *Computer modeling of logical processes. Volume 1. Architecture and languages of the problem solver.*, Fizmatlit, Moscow, 2008, 1024 pp.

[3] Podkolzin A. S., *Computer modeling of logical processes. Volume 2. Experience in training a computer problem solver: logical techniques, set algebra, combinatorics and elementary algebra.*, VINITI RAS, Moscow, 2015, 1153 pp.

[4] Podkolzin A. S., *Computer modeling of logical processes. Volume 3. Experience in teaching computer problem solver: mathematical analysis, differential equations and elementary geometry.*, VINITI RAS, Moscow, 2015, 1320 pp.

[5] Podkolzin A. S., *Computer modeling of logical processes. Volume 4. Experience in teaching computer problem solver: analytical geometry, linear algebra, probability theory, complex analysis and other topics.*, VINITI RAS, Moscow, 2017, 969 pp.

[6] Podkolzin A. S., *Computer modeling of logical processes. Volume 5. Experience in teaching a computer problem solver: elementary physics and chemistry, chess.*, VINITI RAS, Moscow, 2019, 938 pp.

[7] Podkolzin A. S., *Computer modeling of logical processes. Volume 6. Experience in training a computer problem solver: natural language understanding and image analysis.*, VINITI RAS, Moscow, 2019, 757 pp.

[8] Podkolzin A. S., *Computer modeling of logical processes. Volume 7. Automatic creation of logic system techniques: classification of solver techniques, logic assembler, specification compiler, creation of test techniques and refinement of techniques.*, VINITI RAS, Moscow, 2021, 739 pp.

[9] Podkolzin A. S., *Computer modeling of logical processes. Volume 8. Automatic creation of logical system techniques: theorem base, characterization of theorems, creation of technique specifications.*, VINITI RAS, Moscow, 2021, 515 pp.

[10] Podkolzin A. S., *Computer modeling of logical processes. Volume 9. Automatic creation of logical system techniques: logical inference in the theorem base.*, VINITI RAS, Moscow, 2022, 1494 pp.

Автоматизация разметки текстов о жизненных трудностях с использованием больших языковых моделей

А. А. Хлебникова[1], Е. В. Битюцкая[2], Г. В. Калачев[3], Э. Э. Гасанов[4]

Статья посвящена решению проблемы высокой трудоёмкости «ручного» кодирования качественных данных в психологических исследованиях, использующих контент-анализ. Оценивается эффективность методов автоматизированной разметки текстов с применением современных языковых моделей DeepSeek, GPT-4.1 и GPT-4.1-mini и разрабатываются пути повышения точности разметки. Материалом являются описания трудных жизненных ситуаций участников психологического исследования. Исследование подтверждает практическую целесообразность использования языковых моделей в качестве инструмента, значительно сокращающего временны́е затраты исследователя на первичный анализ текстовых данных.

Ключевые слова: контент-анализ, большая языковая модель, GPT-4.1, DeepSeek, трудная жизненная ситуация, копинг (совладание), восприятие ситуации.

[1] *Хлебникова Алёна Андреевна* — аспирант каф. общей психологии ф. психологии, МГУ имени М.В. Ломоносова, e-mail: alena.epochta@gmail.com.

Khlebnikova Alena Andreevna — PhD Student, Department of General Psychology, Faculty of Psychology, Lomonosov Moscow State University.

[2] *Битюцкая Екатерина Владиславовна* — канд. психол. наук, доцент кафедры общей психологии ф. психологии, МГУ имени М.В. Ломоносова, e-mail: bityutskaya_ew@mail.ru.

Bityutskaya Ekaterina Vladislavovna — Candidate of Psychological Sciences, Associate Professor, Department of General Psychology, Faculty of Psychology, Lomonosov Moscow State University.

[3] *Калачев Глеб Вячеславович* — канд. физ.-мат. наук, научный сотрудник кафедры математической теории интеллектуальных систем механико-математического ф. МГУ имени М.В. Ломоносова, e-mail: gleb.kalachev@yandex.ru.

Kalachev Gleb Viacheslavovich — Candidate of Physical and Mathematical Sciences, Researcher, Department of Mathematical Theory of Intellectual Systems, Faculty of Mechanics and Mathematics, Lomonosov Moscow State University.

[4] *Гасанов Эльяр Эльдарович* — д-р физ.-мат. наук, заведующий кафедрой математической теории интеллектуальных систем механико-математического ф. МГУ имени М.В. Ломоносова, e-mail: el_gasanov@mail.ru.

Gasanov Elyar Eldarovich — Doctor of Physical and Mathematical Sciences, Professor, Head of the Department, Department of Mathematical Theory of Intelligent Systems, Faculty of Mechanics and Mathematics, Lomonosov Moscow State University.

Введение

Применение контент-анализа в исследованиях копинга открывает возможности для глубокого анализа феноменологии и смыслов переживания людей. Однако ключевым ограничением метода является высокая трудоёмкость обработки качественных данных. При больших объёмах текстов этап кодирования (разметки) данных часто занимает наибольшее время по сравнению с другими его этапами (сбор данных, разработка кодировочной инструкции, интерпретация результатов). Процесс кодирования «является трудоёмким, дорогим, медленным» [3]. Поэтому с момента начала использования метода в 1960-х годах контент-анализ был тесно связан с разработкой компьютерных технологий для автоматизации кодирования [2].

Компьютерные средства, используемые для контент-анализа в современных исследованиях, можно разделить на программы для автоматической разметки (например, «LIWC», «Leximancer», «DICTION» и др.), программы для ручного и полуавтоматического кодирования (например, «NVivo», «MAXQDA», «ATLAS.ti» и др.) [6] и гибридные инструменты («WordStat», «Quanteda» и др.), сочетающие возможности автоматического кодирования и ручной обработки. Выбор инструмента зависит от объёма данных, типа контент-анализа, необходимости визуализации результатов. В последние годы нейросетевые методы (BERT, GPT и др.) активно применяются для автоматизации разметки [5] и в ряде задач показывают более высокую точность по сравнению с традиционными подходами («LIWC» и др.) [3, 7].

В наших исследованиях копинга и восприятия трудностей разметка используется для контент-анализа описаний трудных жизненных ситуаций, которые получены с помощью открытых вопросов «Методики структурированного описания ситуации» [1, 4]. Вопросы методики и пример случая представлены в Приложении А. Для кодирования описаний привлекаются независимые кодировщики и эксперты. Каждый случай, который используется в исследовании, закодирован двумя кодировщиками и одним экспертом. При этом фрагменты текста, которые закодированы по-разному или вызывают неоднозначные интерпретации, обсуждаются с достижением консенсуса. Кодировочная инструкция разработана совместно Е.В. Битюцкой и Н.Г. Малышевой и включает 187 единиц анализа — категорий и подкатегорий двух видов: относящихся к описанию ситуации в целом (1) и к отдельным вопросам (2). К категориям первого типа относятся эмоции, время, энергия, степень и суть трудности. Категориями второго типа выступают содержание ситуации, копинг, несколько категорий оценки, цели, возможности, ограничения и другие.

Разметка данных связана с выделением смысловых единиц текста описания ситуации, обозначением выделенного фрагмента квадратными скобками и проставлением кода подкатегории. Пример закодированного случая и перечень кодов, использованных в этом примере, представлены в Приложении Б. Размеченные тексты далее обрабатываются с помощью Python-приложения, что позволяет получить таблицу частот категорий и подкатегорий. Такой вариант разметки применялся для контент-анализа, проведённого на выборке 611 описаний трудных жизненных ситуаций (или случаев)[1]. В результате были описаны типы восприятия трудных жизненных ситуаций и проверены предположения о существовании различий между типами [1, 4].

В данной статье мы представляем результаты исследования, целью которого была оценка и повышение эффективности применения языковых моделей для автоматического кодирования текстов. Исследование включало пять последовательных этапов, каждый из которых решал конкретные задачи:

1) оценка базовой эффективности языковых моделей при разметке текстов;

2) оптимизация параметров взаимодействия с языковыми моделями;

3) дообучение GPT-4.1;

4) тестирование эффективности языковых моделей в условиях сокращённой кодировочной инструкции;

5) дообучение GPT-4.1 с применением сокращённой кодировочной инструкции.

1. Задача разметки и метрики качества

Для проведения экспериментов с языковыми моделями было выбрано 100 случаев описаний трудных жизненных ситуаций, которые образовали тестовую выборку. Пример одного из описаний приведён в Приложении А. Далее эти случаи были размечены кодировщиками и экспертом-психологом, и образовали эталонную размеченную выборку. Пример такой разметки приведён в Приложении Б. Тестовая выборка без разметки подавалась на языковую модель, и языковая модель осуществляла разметку

[1] В шестилетнем исследовании (выполнявшемся с ноября 2018 по октябрь 2024 г.) процесс кодирования данных (611 случаев), включая обучение независимых кодировщиков, занял четыре года.

по аналогии с разметкой эксперта-психолога. Полученная автоматическая разметка сравнивалась с эталонной. Сравнение осуществлялось с помощью следующих метрик:

- верные коды (%) — доля точных совпадений кодов и их позиций в тексте автоматической разметки с экспертной разметкой;
- смещённые коды (%) — доля верных кодов, поставленных в неверных фрагментах текста;
- пропущенные коды (%) — доля экспертных кодов, которые языковая модель не обнаружила;
- дубликаты кодов (%) — доля кодов, которые модель поставила избыточно, сверх экспертной разметки;
- лишние коды (%) — доля кодов, которые модель присвоила ошибочно (отсутствуют в экспертной разметке).

Определим более строго структуру документа, который мы размечаем, разметку и процедуру сравнения двух разметок.

Документ представляет собой набор из S *фрагментов*. S — это константа, в нашем случае равная 7 (формулирование трудной жизненной ситуации и ответы на 6 вопросов). Каждый фрагмент — это некоторый текст, который мы представляем как последовательность слов. Число слов в фрагменте f мы называем его *длиной* и обозначаем через $|f|$.

Основная задача состоит в разметке документа, что предполагает разделение каждого фрагмента на смысловые блоки и определение для каждого блока его смысла в терминах фиксированного множества категорий и подкатегорий, обозначаемых *кодами разметки*. Большинство кодов, которые фигурируют в разметке текстов, относятся к кодам подкатегорий. Код ставится в конце смыслового блока, к которому он относится. Во многих случаях встречаются смысловые блоки, закодированные с использованием двух или нескольких кодов. Множество всех кодов разметки обозначим через $\mathcal{L}$.

Роль языковой модели заключается в том, чтобы сделать первичную разметку, которую затем проверяет и корректирует эксперт. Соответственно, качество разметки определяет трудоёмкость проверки и исправления первичной разметки. Разного рода ошибки имеют разную сложность обнаружения и исправления, поэтому мы рассматриваем несколько категорий ошибок. Имея эталонную разметку документа, сделанную экспертом, и тестируемую разметку, сделанную языковой моделью, мы можем вычислить количество ошибок каждого типа.

Для определения необходимых видов ошибок в разметке нам понадобится формальное определение разметки документа. Данное выше

содержательное определение разметки можно формализовать следующим образом.

Разметкой фрагмента f мы называем произвольное множество $T \subseteq \mathbb{N} \times \mathcal{L}$ пар (p, ℓ), где $\ell \in \mathcal{L}$ интерпретируется как код разметки, $p \in \mathbb{N}$ интерпретируется как номер слова, после которого стоит данный код, $p \leq |f|$. *Разметка документа* $T^* = (T^1, \ldots, T^S)$ — набор разметок всех фрагментов этого документа.

Для разметки фрагмента T определим множество $L(T) = \{\ell \mid (p, \ell) \in T\}$, содержащее все метки, входящие в разметку T. Также для кода $\ell \in \mathcal{L}$ определим величину $N(T, \ell) = \big|\{p \in \mathbb{N} \mid (p, \ell) \in T\}\big|$ — количество вхождений кода ℓ в разметку T.

Для разметки документа T^* введём обозначение $N(T^*) = \sum_{i=1}^{S} |T^i|$ — общее число кодов в T^*.

Теперь мы можем определить наши основные метрики сходства тестируемой разметки фрагмента T с эталонной разметкой $\hat{T}$.

1) Количество *верных* кодов $v(\hat{T}, T) = |T \cap \hat{T}|$ — количество точных совпадений с тестируемой и эталонной разметках.

2) Количество *релевантных* кодов

$$r(\hat{T}, T) = \sum_{\ell \in L(\hat{T}) \cap L(T)} \min(N(T, \ell), N(\hat{T}, \ell)).$$

   Оно включает в себя все коды, которые присутствуют и в тестируемой, и в эталонной разметках, но, возможно, на разных позициях.

3) Количество *смещённых* кодов $s(\hat{T}, T) = r(\hat{T}, T) - v(\hat{T}, T)$ — коды, которые присутствуют в обеих разметках, но на разных позициях (не являются верными).

4) Количество *дубликатов*

$$d(\hat{T}, T) = \sum_{\ell \in L(T) \cap L(\hat{T})} \max\big(0, N(T, \ell) - N(\hat{T}, \ell)\big).$$

   Оно включает в себя коды, которые присутствовали в эталонной разметке, но в тестируемой разметке больше вхождений этих кодов.

5) Количество *лишних* кодов

$$e(\hat{T}, T) = \sum_{\ell \in L(T) \setminus L(\hat{T})} N(T, \ell).$$

   Оно включает в себя коды, которые присутствуют в тестируемой разметке, но их нет в эталонной.

6) Количество *пропущенных* кодов

$$m(\hat{T}, T) = \sum_{\ell \in L(\hat{T})} \max\bigl(0, N(\hat{T}, \ell) - N(T, \ell)\bigr).$$

Несложно убедиться, что для любых двух разметок фрагмента $\hat{T}$ и T выполнено

$$v(\hat{T}, T) + s(\hat{T}, T) + m(\hat{T}, T) = |\hat{T}|. \tag{1}$$

Для разметок документа можно определить все эти метрики как сумму соответствующих метрик для фрагментов, то есть метрика $h \in \{v, r, s, d, e, m\}$ для разметок документа определяется как

$$h(\hat{T}^*, T^*) = \sum_{i=1}^{S} h(\hat{T}^i, T^i).$$

Поскольку трудозатраты на исправление разметки мы сравниваем с трудозатратами разметки с нуля, то все посчитанные метрики мы нормируем на общее число кодов в эталонной разметке. В частности, при оценке качества разметки одного документа для метрики $h \in \{v, r, s, d, e, m\}$ мы можем определить нормированную метрику

$$\bar{h}(\hat{T}^*, T^*) = \frac{h(\hat{T}^*, T^*)}{N(\hat{T}^*)}.$$

Если у нас есть выборка $\mathcal{T} = (T_1, \ldots, T_n)$ разметок n документов и набор $\hat{\mathcal{T}} = (\hat{T}_1, \ldots, \hat{T}_n)$ соответствующих эталонных разметок, то мы определяем точно так же, как и для одного документа

$$h(\hat{\mathcal{T}}, \mathcal{T}) = \sum_{i=1}^{n} h(\hat{T}_i, T_i), \qquad \bar{h}(\hat{\mathcal{T}}, \mathcal{T}) = \frac{h(\hat{\mathcal{T}}, \mathcal{T})}{\sum_{i=1}^{n} N(\hat{T}_i)}.$$

В сводных таблицах и при обсуждении экспериментов приводятся именно нормированные метрики качества для всей тестовой выборки.

Учитывая (1), всегда выполнено равенство

$$\bar{v}(\hat{\mathcal{T}}, \mathcal{T}) + \bar{s}(\hat{\mathcal{T}}, \mathcal{T}) + \bar{m}(\hat{\mathcal{T}}, \mathcal{T}) = 1,$$

То есть в сумме нормированные метрики верных, смещённых и пропущенных кодов дают 100%.

Под *суммарным охватом релевантной разметки* в данной статье понимается доля релевантных кодов (сумма долей верных и смещённых кодов), то есть величина $(\bar{v}(\hat{\mathcal{T}}, \mathcal{T}) + \bar{s}(\hat{\mathcal{T}}, \mathcal{T})) \cdot 100\%$. Это связано с тем, что незначительные смещения могут возникать также при сравнении кодирования двух кодировщиков и не являются ошибкой. Трудозатраты эксперта на проверку верных кодов и смещённых кодов значительно ниже, чем на удаление лишних кодов и, в особенности, внесение пропущенных кодов.

2. Программа исследования

Модели. В исследовании были использованы следующие языковые модели:

1) DeepSeek — модель от DeepSeek AI, которая использовалась через веб-интерфейс DeepSeek Chat (https://chat.deepseek.com/). Особенности взаимодействия с моделью:

   - интерфейс чата не позволяет задавать параметры генерации, такие как «температура». Поэтому все эксперименты с данной моделью проводились при стандартных настройках.
   - интерфейс позволяет переключать обычный режим и «режим повышенной точности» DeepThink.

2) GPT-4.1 — модель от OpenAI версии GPT-4.1-2025-04-14, которая использовалась через OpenAI API. API позволяет задавать параметры генерации, в том числе температуру.

3) GPT-4.1-mini — облегчённый вариант GPT-4.1 версии GPT-4.1-mini-2025-04-14, которая также использовалась через OpenAI API.

Коды разметки. Для кодирования текстов использовалось 2 множества кодов:

1) полная кодировочная инструкция I, включающая множество $\mathcal{L}$ из 187 кодов — применялась на этапах 1, 2 и 3;

2) сокращённая кодировочная инструкция I', включающая подмножество $\mathcal{L}' \subseteq \mathcal{L}$ из 54 наиболее релевантных кодов — использовалась на этапах 4 и 5.

Кроме того использовался вспомогательный код «Другое», который здесь для краткости обозначим Λ.

Для удобства определим операцию *стандартной проекции* π, которая удаляет из разметки все коды, не входящие в $\mathcal{L}'$, а также операцию *расширенной проекции* $\tilde{\pi}$, которая заменяет в разметке все коды, не входящие в $\mathcal{L}'$ на Λ. Операции π и $\tilde{\pi}$ определены естественным образом на самих кодах, разметках фрагментов, разметках документов и обучающих/тестовых выборках. Заметим, что для любой разметки T выполнено

$$\pi(\pi(T)) = \pi(T), \qquad \tilde{\pi}(\tilde{\pi}(T)) = \tilde{\pi}(T), \qquad \pi(\tilde{\pi}(T)) = \pi(T).$$

Данные. Материалом исследования послужили описания 210 трудных жизненных ситуаций, полученные с помощью открытых вопросов «Методики структурированного описания ситуации» [1, 4] и размеченные экспертами-психологами.

Таким образом, для каждого случая имеется пара текстов: исходный текст и текст с эталонной разметкой — всего 210 пар текстов. Эти 210 пар текстов разделены на три части: S — обучающая выборка из 100 пар текстов, V — тестовая выборка из 100 пар текстов, а также выборка S_0 из 10 пар текстов, которая использовалась для включения в промпт примеров разметки[2] на этапах 1 и 2.

Также на этапах 4 и 5 использовались множества S', S'_0 и V', которые получены из S, S_0 и V применением расширенной проекции $\tilde{\pi}$, то есть заменой в текстах всех кодов, не входящих в сокращённую кодировочную инструкцию I', на код Λ.

Системное сообщение. В ходе экспериментов использовалось 3 различных варианта системных сообщений:

1) P_1 — включает полную кодировочную инструкцию I;
2) P_{23} — включает полную кодировочную инструкцию I и уточнённые правила разметки;
3) P_{45} — включает сокращённую кодировочную инструкцию I' и уточнённые правила разметки, а также правила использования кода Λ для маркировки смысловых фрагментов, нерелевантных 54 отобранным кодам.

Эксперименты. В каждом эксперименте обучающие примеры либо добавлялись к системному сообщению, либо использовались для дообучения модели GPT-4.1 через OpenAI API со значениями гиперпараметров по умолчанию.

Тестирование эффективности моделей проводилось на тестовой выборке следующим образом: для каждой пары $(s, \hat{t})$ из тестовой выборки модель получала на вход системное сообщение и исходный текст s и размечала его, добавляя в него коды разметки, генерируя на выходе некоторый текст t. При этом даже при строгом запрете на изменение исходного текста иногда модели меняли его, и текст t при обратном удалении разметки не совпадал с исходным текстом s.

В случае, если фрагмент в тексте t после удаления кодов не совпадал с соответствующим фрагментом в тексте s, то каждое такое изменение

[2] В промпт включался только второй элемент каждой пары — текст с эталонной разметкой.

фрагмента помечалось как *изменение исходного текста*. В этом случае применялась библиотека `difflib` из стандартной библиотеки Python для соотнесения позиций в изменённом тексте с позициями в исходном тексте. Далее, используя полученные позиции, формировалась разметка T^*, описанная в разделе 1. В случае, если текст на выходе модели после удаления кодов совпадал с исходным, то разметка T^* формировалась напрямую из позиций кодов в фрагментах текста t.

Размеченный экспертом текст $\hat{t}$ также сравнивался с s и вычислялись позиции всех кодов, формируя эталонную разметку $\hat{T}^*$. После этого вычислялись метрики качества разметки, описанные в разделе 1. Для каждой из метрик считался 95% доверительный интервал методом bootstrap с 2000 перезапусками, и в таблицах эта информация представлена в виде $N \pm \Delta/2$, где N — значение метрики, Δ — длина доверительного интервала. Следует отметить, что доверительные интервалы на самом деле расположены немного несимметрично относительно среднего значения, но сдвинуты незначительно (отношение величины сдвига к длине интервала не выше 0.1).

На этапах 4 и 5 для оценки качества тестируемой разметки $\mathcal{T}$ с помощью сокращённой схемы кодирования с эталонной разметкой $\hat{\mathcal{T}}$ использовались 2 режима вычисления метрик из раздела 1:

1) Сравнение расширенной проекции тестируемой разметки $\tilde{\pi}(\mathcal{T})$ с расширенной проекцией эталонной разметки $\tilde{\pi}(\hat{\mathcal{T}})$. Такой способ включает в сравнение те фрагменты, которым не соответствует ни один код из $\mathcal{L}'$, и которые в данном случае помечаются кодом Λ.

2) Сравнение стандартной проекции тестируемой разметки $\pi(\mathcal{T})$ с проекцией $\pi(\hat{\mathcal{T}})$ эталонной разметки. Такой способ отражает чистое сравнение на подмножестве кодов $\mathcal{L}'$.

Эти методы применялись как для оценки результатов этапов 4 и 5 — там, где разметка сразу производилась множеством кодов $\mathcal{L}' \cup \{\Lambda\}$, так и для разметок, полученных на этапах 2 и 3.

В таблице 1 дано описание этапов экспериментов, в котором указаны модели, системные сообщения, обучающие и тестовые выборки, использованные на каждом этапе.

Далее представлено описание задач и полученных результатов на каждом этапе исследования.

Таблица 1. Этапы экспериментов

Этап	Модели	Промпт	Обучение	Тест
1	DeepSeek, GPT-4.1-mini($t = 0.3$), GPT-4.1($t = 0.3$)	$P_1 + S_0$	нет	V
2	DeepSeek, DeepSeek(DeepThink), GPT-4.1($t = 0$)	$P_{23} + S_0$	нет	V
3	GPT-4.1($t = 0$)	P_{23}	S (3 эпохи)	V
4	DeepSeek, GPT-4.1($t = 0$)	$P_{45} + S'_0$	нет	V'
5	GPT-4.1($t = 0$)	P_{45}	S' (3 эпохи)	V'

3. Результаты и их анализ

Этап 1. Оценка базовой эффективности языковых моделей

Основной задачей первого этапа исследования стала оценка базовой эффективности языковых моделей DeepSeek, GPT-4.1-mini и GPT-4.1. На этом этапе был создан единый промпт с полной кодировочной инструкцией (187 кодов), правилами разметки и примерами размеченных кодировщиками и экспертом десяти текстов-описаний трудных жизненных ситуаций. Эти десять случаев не входили в тестовую выборку.

Результаты первого этапа представлены в таблице 2. Языковая модель DeepSeek продемонстрировала точность 34% верных кодов, однако пропустила 57% экспертных кодов. Суммарный охват релевантной разметки составил 43%. Языковая модель показала умеренный уровень избыточного кодирования (4% дубликатов кодов) и допустила 26% ошибочных присвоений кодов смысловым единицам текста.

GPT-4.1 показала схожий результат по основным метрикам (32% верных кодов, 57% пропусков), но продемонстрировала более высокую склонность к генерации ошибочных кодов (42% лишних кодов) при меньшем уровне дублирования (2%). Общий охват релевантной разметки (43%) подтверждает конкурентоспособность данной модели в задачах автоматического кодирования.

Наименее эффективной оказалась модель GPT-4.1-mini с показателем 14% верных кодов и 76% пропусков. При полном отсутствии дубликатов кодов модель допустила 22% ошибочных кодов, а суммарный охват релевантной разметки составил лишь 24%. Ключевым недостатком данной языковой модели стало неприемлемое количество изменений исходного

Таблица 2. Результаты первого этапа

Тип кодов, %	DeepSeek	GPT-4.1	GPT-4.1-mini
Верные	34.4 ± 2.8	32.3 ± 3.1	13.6 ± 1.9
Смещённые	8.5 ± 1.2	10.8 ± 1.6	10.6 ± 1.3
Пропущенные	57.1 ± 2.2	56.9 ± 3.1	75.8 ± 2.1
Дубликаты	3.6 ± 1.4	1.6 ± 0.6	0.0 ± 0.0
Лишние	25.6 ± 3.1	42.5 ± 4.8	21.7 ± 2.3

Примечание: здесь и далее в таблицах представлены доли всех типов кодов относительно экспертной разметки и 95% доверительный интервал.

текста (328 правок с сильным искажением исходного текста), что полностью исключает её использование в рамках задач автоматического кодирования. У остальных моделей количество внесённых изменений было незначительным (в среднем 4 правки с незначительным изменением исходного текста).

Полученные результаты позволили отобрать две наиболее перспективные модели — DeepSeek и GPT-4.1 — для дальнейшей оптимизации.

Этап 2. Оптимизация параметров взаимодействия с языковыми моделями

Основной задачей второго этапа исследования стала целенаправленная оптимизация параметров взаимодействия с языковыми моделями DeepSeek и GPT-4.1, продемонстрировавшими лучшие результаты на предыдущем этапе. По аналогии с первым этапом в промпт были включены десять случаев эталонной разметки. Меры оптимизации включали уточнение формулировки промпта, правил разметки и настройку параметра генерации «температура» ($t = 0$) для модели GPT-4.1. Снижение значения данного параметра до нуля было призвано минимизировать стохастичность выходных данных модели и максимизировать детерминированность её ответов. Параллельно проводилось тестирование режима повышенной точности DeepThink модели DeepSeek для сравнения эффективности различных подходов.

Результаты демонстрируют более высокие показатели эффективности языковых моделей по ключевым метрикам (таблица 3). Модель GPT-4.1 показала наиболее значительный прогресс: доля верных кодов возросла с 32% до 46%, а уровень пропусков снизился с 57% до 46%. Суммарный охват релевантной разметки достиг 54%. При этом отмечается увеличение доли лишних кодов с 42% до 63%, что свидетельствует о возросшей ак-

тивности языковой модели в присвоении кодов, часть которых не соответствует экспертной разметке. Показатель дублирования кодов увеличился с 2% до 8%, указывая на тенденцию к избыточному кодированию.

Языковая модель DeepSeek также показала улучшение результатов: точность верных кодов повысилась с 34% до 36%, а доля пропусков незначительно снизилась (с 57% до 56%). Суммарный охват релевантной разметки составил 45%. Количество лишних кодов осталось на сопоставимом уровне (27% против 26% на первом этапе), в то время как показатель дублирования увеличился с 4% до 5%.

Активация режима повышенной точности DeepThink языковой модели DeepSeek существенно не изменила показатели точности. Суммарный охват релевантной разметки составил 45% (37% верных кодов) при 55% пропусков. Модель продемонстрировала аналогичный уровень дублирования (5%) и незначительное снижение доли лишних кодов (26%).

Сравнительный анализ результатов двух языковых моделей выявляет различные стратегии поведения после оптимизации: GPT-4.1 продемонстрировала тенденцию к экспансии кодирования, характеризующуюся значительным увеличением количества присваиваемых кодов, что выразилось в возрастании числа как корректных, так и ошибочных кодов. В противоположность этому, модель DeepSeek сохранила консервативную стратегию кодирования, проявляющуюся в минимальном приросте ошибочных идентификаций кодов.

Такое различие в стратегиях может объясняться свойствами моделей и разной чувствительностью к оптимизации их параметров. Экспансивное поведение GPT-4.1 свидетельствует о её большей восприимчивости к изменениям промпта и параметров генерации, в то время как консервативность DeepSeek может указывать на бóльшую устойчивость к изменениям.

Полученные результаты подтверждают эффективность применяемых процедур оптимизации и обосновывают целесообразность их использования для повышения эффективности автоматизированного кодирования текстовых данных.

Этап 3. Дообучение модели GPT-4.1

Третий этап исследования был посвящён реализации более сложного подхода к автоматизации — проведению процедуры дообучения (fine-tuning) языковой модели GPT-4.1. Данный метод подразумевает адаптацию внутренних весов модели под специфику конкретной задачи. Для обучения использовался набор данных, состоящий из 100 новых эталонных примеров экспертной разметки, преобразованных в 100 текстовых пар формата

«исходный текст – текст с разметкой». Результаты представлены в таблице 3.

Языковая модель GPT-4.1 продемонстрировала 49% верных кодов при значительном снижении доли пропусков до 42%. Суммарный охват релевантной разметки составил 59%. Показатель дублирования кодов остался на умеренном уровне (8%), а количество лишних кодов составило 48%. Эти результаты свидетельствуют о высокой эффективности подхода с дообучением для задач автоматизированного кодирования.

Таблица 3. Результаты второго и третьего этапов

	Этап 2			Этап 3
Тип кодов, %	DeepSeek	DeepSeek (DeepThink)	GPT-4.1	GPT-4.1
Верные	36.2 ± 2.9	36.6 ± 2.6	46.1 ± 2.7	48.5 ± 2.7
Смещённые	8.3 ± 1.2	8.0 ± 1.3	8.3 ± 1.2	10.0 ± 1.5
Пропущенные	55.5 ± 2.5	55.4 ± 2.4	45.7 ± 2.4	41.5 ± 2.7
Дубликаты	4.6 ± 1.4	4.9 ± 1.4	8.3 ± 2.0	7.5 ± 1.7
Лишние	27.0 ± 3.2	26.4 ± 3.0	63.0 ± 6.5	47.7 ± 3.7

Этап 4. Применение сокращённой схемы кодирования

Четвёртый этап исследования был направлен на снижение сложности кодировочной инструкции, что, согласно нашему предположению, должно было повысить точность автоматизированной разметки. При этом мы исходили из результатов предыдущих исследований [4], которые показали, что для решения задачи классификации значимы лишь определённые коды, частота встречаемости которых в разметке позволяет выделять типы восприятия жизненных трудностей. Изначально мы предположили, что если в промпте останутся наиболее значимые для классификации коды (а коды, которые не используются для разделения на типы, применяться не будут), то это упрощение создаст условия для улучшения точности разметки.

В рамках данного этапа была выполнена сравнительная оценка эффективности моделей DeepSeek и GPT-4.1 с использованием промпта, адаптированного под сокращённую схему кодирования, включающую 54 наиболее релевантных кода. Подготовка обучающего материала состояла в том, что из исходного материала удалялись коды, которые не входили в множество выделенных 54 кодов. Вместо удалённых кодов вносился специальный код «Другое». Аналогичная замена кодов, не входящих в сокращённую схему, на код «Другое» была произведена и в инструкции для промпта. Если говорить формально, то к обучающему материалу

и к инструкции для промпта была применена операция расширенной проекции, которая описана в разделе 2.

Таблица 4. Результаты четвёртого и пятого этапов с учётом кода «Другое»

Тип кодов, %	Этап 4		Этап 5
	DeepSeek	GPT-4.1	GPT-4.1
Верные	45.6 ± 3.0	41.3 ± 3.0	49.2 ± 2.7
Смещённые	6.6 ± 1.3	8.3 ± 1.8	11.3 ± 1.7
Пропущенные	47.8 ± 2.8	50.4 ± 3.1	39.5 ± 2.4
Дубликаты	12.7 ± 2.9	9.6 ± 2.5	14.4 ± 3.3
Лишние	21.5 ± 2.9	64.1 ± 6.9	31.6 ± 3.5

Результаты четвёртого этапа, когда сравнение ведётся с учётом кода «Другое», представлены в таблице 4. Модель DeepSeek показала точность 46% верных кодов, 7% смещенных кодов при 48% пропусков. Количество лишних кодов снизилось до 22%, а показатель дублирования составил 13%. Суммарный охват релевантной разметки достиг 52%. Тем самым при сокращённой схеме DeepSeek показывает лучшие результаты по сравнению с результатами этапа 2.

Модель GPT-4.1 при сокращённой схеме демонстрирует ухудшение результатов по сравнению с этапом 2, а именно 41% верных кодов, 8% смещенных кодов при 50% пропусков. Количество лишних кодов составило 64%, а показатель дублирования — 10%. Суммарный охват релевантной разметки составил 50%.

Отметим, что доля кода «Другое» в экспертной разметке оказалась равна 46% от всех кодов, внесённых в тексты. При таком условии большое влияние на результаты может оказывать эффект угадывания кода «Другое». Чтобы устранить влияние этого кода на точность разметки, мы выполнили оценку точности разметки без учёта кода «Другое». Если говорить более формально, к результатам этапа 4 была применена операция стандартной проекции. Результаты этого сравнения приведены в таблице 5.

Заметим, что полученные результаты неоднозначны. Для модели DeepSeek мы ожидаемо видим уменьшение доли верных кодов до 28% (показатель релевантной разметки 34%). При этом возросла доля пропущенных кодов до 66%. Для модели GPT-4.1 мы неожиданно наблюдаем увеличение доли верных кодов до 47%, а доли смещённых кодов до 9% при уменьшении доли пропущенных кодов до 44%. Можно заметить, что эти результаты лучше, чем результаты этапа 2, но при этом доля лишних кодов становится слишком высокой — 112%. Можно предположить, что

Таблица 5. Результаты четвёртого и пятого этапов без учёта кода «Другое»

Тип кодов, %	Этап 4		Этап 5
	DeepSeek	GPT-4.1	GPT-4.1
Верные	28.1 ± 3.6	47.0 ± 3.7	30.3 ± 3.6
Смещённые	5.7 ± 1.5	9.1 ± 2.1	9.4 ± 2.1
Пропущенные	66.2 ± 3.5	44.4 ± 3.7	60.3 ± 3.2
Дубликаты	6.5 ± 3.0	11.5 ± 3.7	3.3 ± 2.0
Лишние	24.6 ± 4.9	112.3 ± 16.2	35.5 ± 5.5

языковая модель DeepSeek лучше приспособилась к угадыванию кода «Другое», тогда как модель GPT-4.1 показывает лучшие результаты при разметке релевантных кодов, но при этом избыточно добавляет лишние коды.

Чтобы понять, насколько на обучаемость моделей влияет именно сокращение числа кодов, были выполнены следующие действия. К результатам этапа 2 была применена операция расширенной и стандартной проекции. Т.е. в разметке, полученной на этапе 2, и в экспертной разметке все коды, не входящие в сокращённую схему, были заменены на код «Другое» в случае расширенной проекции, и удалены коды, не входящие в сокращённую схему — в случае стандартной проекции. Результаты сравнения расширенной и стандартной проекций представлены в таблицах 6 и 7.

Таблица 6. Сравнение расширенных проекций второго и третьего этапов на сокращённое множество кодов

Тип кодов, %	Разметка этапа 2		Разметка этапа 3
	DeepSeek	GPT-4.1	GPT-4.1
Верные	48.6 ± 3.7	52.6 ± 3.0	60.6 ± 3.1
Смещённые	8.3 ± 1.5	7.7 ± 1.6	8.6 ± 1.7
Пропущенные	43.1 ± 3.0	39.7 ± 2.7	30.8 ± 2.8
Дубликаты	10.8 ± 2.7	12.0 ± 2.8	17.5 ± 2.7
Лишние	22.5 ± 2.8	37.9 ± 4.0	29.5 ± 3.2

Анализируя полученные результаты, мы можем видеть, что как для модели DeepSeek, так и для модели GPT-4.1 результаты расширенной проекции этапа 2 лучше, чем результаты этапа 4. В то же время для

Таблица 7. Сравнение стандартных проекций второго и третьего этапов на сокращённое множество кодов

Тип кодов, %	Разметка этапа 2		Разметка этапа 3
	DeepSeek	GPT-4.1	GPT-4.1
Верные	33.6 ± 4.1	42.8 ± 3.5	43.7 ± 3.7
Смещённые	8.4 ± 2.1	7.2 ± 1.6	8.8 ± 2.1
Пропущенные	58.0 ± 3.4	50.0 ± 3.5	47.5 ± 3.8
Дубликаты	5.1 ± 2.7	9.2 ± 3.8	6.9 ± 3.0
Лишние	24.4 ± 4.7	56.3 ± 8.7	35.7 ± 4.7

стандартной проекции ситуация противоречивая: сокращённая схема даёт для модели GPT-4.1 преимущество, а для модели DeepSeek — ухудшение результатов. Таким образом, сокращённая схема не улучшает показатели эффективности разметки языковых моделей: они с примерно одинаковыми показателями могут обрабатывать как полное, так и сокращённое множество кодов.

Этап 5. Дообучение модели GPT-4.1 с применением сокращённой схемы кодирования

Пятый этап исследования был направлен на оценку влияния процедуры дообучения (fine-tuning) языковой модели GPT-4.1 на точность автоматизированного кодирования в условиях применения сокращённой схемы кодирования. Этап предусматривал оценку эффективности дообученной версии модели GPT-4.1 с использованием оптимизированного промпта, адаптированного под сокращённую схему кодирования (54 кода). Для обучения использовался тот же, что и на этапе 4, набор из 100 текстовых пар формата «исходный текст — текст с разметкой», в разметке которых были оставлены только коды из сокращённой схемы, а коды, не входящие в сокращённую схему, были заменены на код «Другое». Как и на предыдущем этапе, инструкция для промпта включала специализированный код «Другое» (для маркировки смысловых фрагментов, нерелевантных 54 отобранным кодам). Мы предположили, что явное задание правил обработки релевантных и нерелевантных смысловых элементов текста в сочетании с предварительным дообучением модели будет способствовать повышению эффективности разметки.

Результаты пятого этапа представлены в таблицах 4 и 5.

Применение дообученной модели GPT-4.1 с промптом, включающим код «Другое», позволило достичь высоких результатов. Модель показала

точность 49% верных кодов при доле пропусков 40%. Суммарный охват релевантной разметки составил 61%. Количество лишних кодов осталось на умеренном уровне (32%), а показатель дублирования увеличился до 14%, что свидетельствует о возросшей активности модели при сохранении высокой точности. Наблюдаемый рост доли смещённых кодов (11%) указывает на незначительные ошибки в определении границ смысловых единиц, которые, однако, не снижают общей эффективности. В то же время, если сравнивать результаты этапа 3 (показатель релевантной разметки 59%) и этапа 5 без учета кода «Другое» (40%), то можно наблюдать ухудшение качества получаемой разметки. Исходя из этого, можно сделать вывод, что модель GPT-4.1 научается правильно определять код «Другое», но содержательные коды различает хуже.

Сравнение результатов дообучения по сокращённой схеме (54 кода) с проекциями дообученной полной схемы (см. таблицы 6 и 7) показывает, что и стандартная, и расширенная проекции результатов этапа 3 дают лучшую разметку, чем дообученная модель этапа 5. Это позволяет сделать вывод о том, что использование сокращённой схемы не даёт улучшения точности разметки по сравнению с полной схемой при использовании языковых моделей.

4. Заключение

Проведённое исследование демонстрирует принципиальную возможность и практическую эффективность использования современных языковых моделей GPT-4.1 и DeepSeek для автоматизации трудоёмкого процесса кодирования текстовых данных в психологических исследованиях. На основе сравнительного анализа пяти последовательных этапов работы с моделями DeepSeek и GPT-4.1 были получены следующие выводы.

В задачах автоматического кодирования языковые модели демонстрируют различную эффективность. GPT-4.1 легче поддается оптимизации и дообучению, в то время как DeepSeek проявила стратегию кодирования с меньшим количеством ошибок, но и с меньшей полнотой охвата релевантной разметки (45%).

Оптимизация промптов и параметров генерации существенно улучшает качество разметки (GPT-4.1). Снижение параметра «температура» до нуля и уточнение формулировок правил кодирования позволили повысить точность релевантной разметки для GPT-4.1 с 43% до 54% на втором этапе исследования.

Дообучение на релевантных данных является эффективным методом повышения точности разметки. Процедура fine-tuning модели GPT-4.1 на 100 размеченных экспертами случаях позволила достичь наиболее высоких показателей на третьем этапе: 59% релевантных кодов.

Использование сокращённой схемы кодирования (54 кода) в целом не даёт преимуществ перед использованием полной схемы (187 кодов) для автоматизированной разметки с применением языковых моделей.

Полученные результаты обосновывают целесообразность использования языковых моделей в качестве инструмента, предназначенного для первичного анализа больших массивов текстовых данных, что значительно сокращает временны́е затраты исследователя. Говоря о первичном анализе, мы утверждаем, что важным условием применения языковых моделей для разметки данных является дальнейшая работа с текстами обученных кодировщиков и экспертов.

Перспективы исследования. Неожиданным итогом данного исследования являются схожие результаты эффективности разметки при использовании полной и сокращенной схемы кодирования (187 кодов и 54 кода). На первый взгляд, сокращение множества кодов должно было улучшить показатели языковой модели. Однако этот исследовательский ход в целом не дал преимуществ качества автоматизированной разметки. Одно из возможных объяснений такого эффекта связано с семантической или смысловой полнотой 187-элементного множества кодов. То есть данный набор кодов обеспечивает хорошую степень смыслового охвата (или «покрытия» текста кодами и стоящими за ними смысловыми единицами), тогда как результат кодирования на основе сокращенного множества кодов зияет смысловыми пробелами. Если принять гипотезу о семантической полноте, то в последующих исследованиях языковые модели можно использовать как инструмент измерения семантической полноты множеств кодов для контент-анализа текстов.

Предложенный подход может быть адаптирован для различных задач контент-анализа в психологии и смежных дисциплинах.

5. Благодарности

Благодарим доцента факультета психологии МГУ имени М.В. Ломоносова Н.Г. Малышеву за сотрудничество при разработке кодировочной инструкции для контент-анализа и аспирантку этого факультета А.Г. Докучаеву за помощь в обработке данных и экспертную работу. Выражаем признательность рецензентам статьи за ценные рекомендации.

Финансирование. Исследование выполнено за счет гранта Российского научного фонда № 25-18-00737, https://rscf.ru/project/25-18-00737/

Список литературы

[1] Е. В. Битюцкая, М. И. Кунашенко, “Стремление к трудности как тип восприятия жизненных ситуаций”, *Вестник Московского университета. Серия 14: Психология*, **47**:1 (2024), 56–87.

[2] Н. Н. Богомолова, Н. Г. Малышева, Т. Г. Стефаненко, “Контент-анализ”, *Социальная психология: практикум: учеб. пособие для студентов вузов*, ред. Т. В. Феломеева, Аспект Пресс, М., 2009, 131–162.

[3] J. Biggiogera, G. Boateng, P. Hilpert *et al.*, *BERT meets LIWC: Exploring State-of-the-Art Language Models for Predicting Communication Behavior in Couples' Conflict Interactions*, 2021, arXiv: 2106.01536.

[4] E. V. Bityutskaya, E. E. Gasanov, K. V. Khazova, N. A. Patrashkin, “Classifying the Perception of Difficult Life Tasks: Machine Learning and/or Modeling of Logical Processes”, *Psychology in Russia: State of the Art*, **17**:2 (2024), 64–84.

[5] T. B. Brown, B. Mann, N. Ryder, N. Subbiah *et al.*, *Language Models are Few-Shot Learners*, 2020, arXiv: 2005.14165.

[6] K. Krippendorff, *Content Analysis: An Introduction to Its Methodology*, 4th ed., Sage Publications, Inc., Thousand Oaks, CA, 2019.

[7] L. Tavabi, T. Tran, K. Stefanov *et al.*, “Analysis of Behavior Classification in Motivational Interviewing”, Proceedings of the Conference on Computational Linguistics and Clinical Psychology (CLPsych), 2021, 110–115.

Приложение А. Пример исходного описания трудной жизненной ситуации

Пример случая (мужчина, 38 лет). Курсивом даны инструкция и вопросы Методики структурированного описания ситуации.

Сформулируйте свою жизненную ситуацию, которая является трудной задачей, требующей решения в данный период времени.
Трудная финансовая ситуация. Высокая цена съёмной квартиры, больше половины зарплаты уходит на оплату аренды, плюс оплата кредита, плюс задолженность на кредитной карте. Естественно, ситуация накладывает отпечаток и на состояние.

1. Как Вы её воспринимаете, оцениваете, переживаете и преодолеваете? (Какие действия помогают вам преодолеть ситуацию или свое

состояние).
Присутствует недовольство, небольшой элемент угнетённости, невозможность в данный момент позволить себе то, что хочется, создает удручающие ощущения. Это произошло из-за нерационального распределения денег, желания позволить себе и своей семье больше. Небольшие накопления, практически все, ушли на то, чтобы погасить задолженность по кредитной карте. Помочь решить эту ситуацию может четко выверенный план. Составлен план по выходу из кризисной ситуации.

2. Каковы Ваши цели в этой ситуации?
Переезд в квартиру вдвое дешевле, полный подсчет необходимых расходов на жизнь, постоянные накопления для создания неприкасаемого запаса на форс-мажорные случаи.

3. Какие возможности и ограничения есть у Вас при достижении цели?
Возможности – зарплата. Ограничения – зарплата. Когда у меня получится найти дополнительный доход, будет проще преодолевать ситуацию.
4. Нужна ли Вам в этой ситуации помощь (поддержка) окружающих людей?
Да. Прежде всего моей семьи, моей жены. Вместе мы составили план и собираемся ему следовать. А если у неё получится найти работу, ту, которая ей понравится и будет приносить доход, будет еще лучше.

5. Если всё сложится очень плохо, то что это будет? (Максимальный неуспех).
Очередной уход в долговую яму, потеря единомыслия со своей супругой.

6. Опишите, что для Вас будет максимально успешным выходом, разрешением ситуации?
В данный момент – жизнь по средствам, выход из финансового кризиса, увеличение совокупного дохода семьи, накопление средств.

Приложение Б. Пример эталонной разметки, выполненной экспертом-психологом

Пример случая (мужчина, 38 лет). Курсивом даны инструкция и вопросы Методики структурированного описания ситуации. Коды разметки обозначены звёздочками.

Сформулируйте свою жизненную ситуацию, которая является трудной задачей, требующей решения в данный период времени.
[Трудная финансовая ситуация. *Ф1.1] [Высокая цена съёмной квартиры, больше половины зарплаты уходит на оплату аренды, плюс оплата кредита, плюс задолженность на кредитной карте. *Ф1.1.П] [Естественно, ситуация накладывает отпечаток и на состояние. *С8]

1. Как Вы её воспринимаете, оцениваете, переживаете и преодолеваете? (Какие действия помогают вам преодолеть ситуацию или свое состояние).
[Присутствует недовольство, *A4] [небольшой элемент угнетённости, *E2] [невозможность в данный момент позволить себе то, что хочется, *1K2, *Б1] [создает удручающие ощущения. *A3] [Это произошло из-за нерационального распределения денег, желания позволить себе и своей семье больше. Небольшие накопления, практически все, ушли на то, чтобы погасить задолженность по кредитной карте. *1B5, *1D7, *G1] [Помочь решить эту ситуацию может четко выверенный план. *1D4, *1D17] [Составлен план по выходу из кризисной ситуации. *1D4]

2. Каковы Ваши цели в этой ситуации?
[Переезд в квартиру вдвое дешевле, *2A2] [полный подсчет необходимых расходов на жизнь, *2A2] [постоянные накопления для создания неприкасаемого запаса на форс-мажорные случаи. *2A5]

3. Какие возможности и ограничения есть у Вас при достижении цели?
[Возможности – зарплата. *3A11] [Ограничения – зарплата. *3B9] [Когда у меня получится найти дополнительный доход, будет проще преодолевать ситуацию. *3A5]

4. Нужна ли Вам в этой ситуации помощь (поддержка) окружающих людей?
[Да. Прежде всего моей семьи, моей жены. *4A3] [Вместе мы составили план и собираемся ему следовать. *4B4] [А если у неё получится найти работу, ту, которая ей понравится и будет приносить доход, будет еще лучше. *4B5]

5. Если всё сложится очень плохо, то что это будет? (Максимальный неуспех).
[Очередной уход в долговую яму, *5B2, *G2] [потеря единомыслия со своей супругой. *5B2]

6. Опишите, что для Вас будет максимально успешным выходом, разрешением ситуации?
[В данный момент – жизнь по средствам, *6B4, *Б1] [выход из финансового кризиса, *6B2] [увеличение совокупного дохода семьи, накопление средств. *6B1]

Перечень применённых кодов

В таблице Б.1 приведён перечень кодов, которые использовались в разметке случая (в порядке их появления в тексте). Коды, которые начинаются с букв, относятся к описанию ситуации в целом, с цифр — к отдельным вопросам (первая цифра кода соответствует номеру вопроса).

Таблица Б.1. Коды, применявшиеся при разметке случая

Код	Подкатегория	Категория
Ф1.1	Материальная трудность	Жизненная сфера
Ф1.1.П	Подробности о материальной трудности	Жизненная сфера
C8	Собственное состояние и совладание с ним	Суть трудности
A4	Негативные неинтенсивные	Эмоции
E2	Низкий уровень энергии	Энергия
1K2	Неподконтрольность ситуации	Критерии оценки
Б1	Упоминание времени	Время
A3	Негативные интенсивные	Эмоции
1B5	Аргументы оценки	Основания оценки
1D7	Анализ опыта	Копинг
G1	Развитие ситуации	Динамика
1D4	Планомерный копинг	Копинг
1D17	Оценка действенности копинга	Копинг
2A2	Приближение (когнитивное фокусирование на трудной задаче)	Направленность цели
2A5	Развитие, увеличение	Направленность цели
3A11	Материальные возможности	Возможности
3B9	Материальные ограничения	Ограничения
3A5	Необходимость активности	Возможности
4A3	Уверенность в необходимости помощи	Необходимость помощи
4B4	Взаимодействие	Содержание помощи
4B5	Инструментальная помощь	Содержание помощи
5B2	Утрата чего-либо	Содержание неуспеха
G2	Повторяемость, цикличность	Динамика
6B4	Поддержание существующего положения	Содержание успеха
6B2	Избавление от чего-либо	Содержание успеха
6B1	Появление чего-то нового	Содержание успеха

The Use of Language Models in Automated Markup of Texts on Life Difficulties

Khlebnikova A.A., Bityutskaya E.V., Kalachev G.V., Gasanov E.E.

The paper addresses the laborious nature of manual coding of qualitative data in psychological studies that use content analysis. The effectiveness of automated text markup methods utilizing modern language models such as DeepSeek, GPT-4.1, and GPT-4.1-mini is assessed, and approaches to improve markup accuracy are developed. The work is based on descriptions of difficult life situations experienced by participants in a psychological study. The study confirms the practical feasibility of using language models as a tool that significantly reduces the time spent by researchers on the initial analysis of text data.

Keywords: content analysis, large language model, GPT-4.1, DeepSeek, difficult life situation, coping, situation perception.

References

[1] E. V. Bityutskaya, M. I. Kunashenko, "Striving for Difficulty as a Type of Perception of Life Situations", *Lomonosov Psychology Journal*, **47**:1 (2024), 56–87 (In Russian).

[2] N. N. Bogomolova, N. G. Malysheva, T. G. Stefanenko, "Content Analysis", *Social Psychology: Practicum: A Textbook for University Students*, ed. T. V. Folomeeva, Aspect Press, Moscow, 2009, 131–162 (In Russian).

[3] J. Biggiogera, G. Boateng, P. Hilpert *et al.*, *BERT meets LIWC: Exploring State-of-the-Art Language Models for Predicting Communication Behavior in Couples' Conflict Interactions*, 2021, arXiv: 2106.01536.

[4] E. V. Bityutskaya, E. E. Gasanov, K. V. Khazova, N. A. Patrashkin, "Classifying the Perception of Difficult Life Tasks: Machine Learning and/or Modeling of Logical Processes", *Psychology in Russia: State of the Art*, **17**:2 (2024), 64–84.

[5] T. B. Brown, B. Mann, N. Ryder, N. Subbiah *et al.*, *Language Models are Few-Shot Learners*, 2020, arXiv: 2005.14165.

[6] K. Krippendorff, *Content Analysis: An Introduction to Its Methodology*, 4th ed., Sage Publications, Inc., Thousand Oaks, CA, 2019.

[7] L. Tavabi, T. Tran, K. Stefanov *et al.*, "Analysis of Behavior Classification in Motivational Interviewing", Proceedings of the Conference on Computational Linguistics and Clinical Psychology (CLPsych), 2021, 110–115.

Часть 2
Специальные вопросы теории интеллектуальных систем

Адаптивно регуляризованный псевдообратный префильтр для передачи сигнала в многоантенных системах радиосвязи

Е. А. Бобров[1] , Д. С. Миненков[2] , Д. А. Юдаков[3]

Современные сотовые сети используют технологию передачи сигнала с множеством антенн на стороне базовой станции и пользователя. В работе исследуется адаптивный префильтр, использующий особую регуляризацию на основе сингулярного разложения матриц. Проводится теоретический анализ, оценка производительности и сравнение с другими методами на симуляциях в модели канала "Quadriga".

Ключевые слова: Телекоммуникации, технология MIMO, оптимизация, сингулярное разложение, отношение сигнал-интерференция-шум, спектральная эффективность

1. Введение

В системах радиосвязи с множеством передающих и принимающих антенн (multiple-input multiple-output, MIMO) префильтр[4] является важной частью обработки сигнала на нисходящей линии связи, поскольку позволяет фокусировать энергию передаваемого сигнала на меньших областях (именно там, где располагаются приемники), обеспечивая большую спектральную эффективность при меньшей передаваемой мощности [1, 2].

[1] *Бобров Евгений Александрович* — к.ф.-м.н. (2025), аспирантура кафедры Математических Методов Прогнозирования факультета ВМК МГУ, e-mail: eugenbobrov@ya.ru.

Bobrov Evgeny Aleksandrovich — PhD, Lomonosov Moscow State University, Faculty of Computational Mathematics and Cybernetics, Department of Mathematical Methods of Forecasting.

[2] *Миненков Дмитрий Сергеевич* — н.с. мех.-мат. ф-та МГУ; старший преподаватель КНТ МФТИ, e-mail: minenkov.ds@gmail.com.

Minenkov Dmitry Sergeevich — research fellow, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics; senior teacher, Moscow Institute of Physics and Technology, Department of Mathematics and Mathematical Methods in Physics

[3] *Юдаков Даниил Андреевич* — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: d.yudakov43@gmail.com.

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

[4] Линейный префильр – это линейное преобразование вектора передаваемых символов, производимое на передатчике. В англоязычной литературе встречается под разными названиями, в т.ч. Precoding, Beamforming, Transmitting filter, etc.

Различные линейные префильтры позволяют либо максимизировать подачу энергии к пользователю, как в случае передачи с сопряжённым префильтром (далее – СФ, в англоязычной литературе известен как Maximum Ratio Transmission, MRT), либо полностью устранять межпользовательские помехи [5], как в методе псевдообратного префильтра (далее – ПФ, в англоязычной литературе – Zero-Forcing, ZF) [3].

В случае алгоритмов регуляризованного псевдообратного префильтра (РПФ, Regularized Zero-Forcing, RZF) достигается баланс между максимизацией мощности сигнала и минимизацией межпользовательской интерференции [4, 5, 6, 7, 8], при этом сохраняется относительно низкая вычислительная сложность по сравнению с нелинейными префильтрами. Обычно рассматривается "скалярная" регуляризация, то есть регуляризация с помощью скалярного множителя при единичной матрице соответствующего размера. В данной работе исследуется явная эвристическая формула для диагональной регуляризации, которая даёт лучшие результаты по сравнению со скалярным РПФ при той же асимптотической сложности алгоритма. Исследование подкреплено теоретическим обоснованием и тестированием в симуляторе (программном пакете) "Quadriga" [9].

Существуют и другие подходы построения линейного префильтра, такие как префильтр с учётом постфильтра (detection-aware precoding) [10], блочно-диагональный префильтр [11], а также префильтры с распределённой мощностью [12]. Также существуют различные нелинейные методы построения префильтра, такие как кодирование с учётом известного помехового сигнала (Dirty Paper Coding, DPC) и векторная перестановка (Vector Perturbation, VP), однако они значительно сложнее в реализации [13], и при использовании множества антенн предпочтение отдается линейным методам префильтра.

Существуют качественные обзоры методов префильтра [14, 15], а также статьи, рассматривающие различные аспекты и модификации этих алгоритмов (см., например, [16, 17, 18]). В частности, в работе [18] представлены различные варианты РПФ в случае нескольких базовых станций, позволяющие уменьшить межсотовые помехи.

Большинство работ, для упрощения анализа, не уделяют должного внимания пользователям с несколькими антеннами. В нашей работе рассматривается одна базовая станция (с большим количеством передающих антенн), обслуживающая пользователей также с несколькими антеннами каждый, при этом каждому пользователю передаётся меньшее число каналов данных, чем число антенн. Это объясняется тем, что на

[5] Помехи на приемнике, возникающие в результате одновременной передачи сигналов разным пользователям, в литературе часто называется "интерференцией". Составляет значительную (часто бОльшую) часть всех помех.

практике каналы между различными антеннами одного пользователя часто пространственно коррелированы [19]. В результате матрицы каналов пользователя плохо обусловлены (или даже вырождены), и невозможно эффективно передавать данные с помощью максимального числа потоков.

Для решения этой проблемы вместо полной матрицы канала пользователя в префильтре можно использовать векторы из её сингулярного разложения, соответствующие наибольшим сингулярным значениям [20]. В случае пользователя с одной антенной канал можно нормализовать [6], и коэффициенты нормализации соответствуют потерям мощности сигнала на пути от передатчика к приемнику (pathloss, PL), которые могут отличаться на несколько порядков: типичные значения мощностей полученного сигнала (Reference Signal Received Power, RSRP) варьируются от -130 дБм до -70 дБм. При использовании сингулярного разложения, сингулярные значения разных пользователей имеют тот же порядок, что и соответствующие потери мощности на пути, и также сильно варьируются.

Нам удалось найти простую эвристическую формулу, которая превосходит известные алгоритмы РПФ. Алгоритмы типа РПФ [4] используют скалярную регуляризацию вида $\lambda\mathbf{I}$. В работе Э. Бьёрнсона [5] доказано из общих соображений, что максимизация функции отношения сигнал-интерференция-шум (ОСИШ[6] , Signal-to-Noise-plus-Interference Ratio, SINR), включая рассматриваемую суммарную спектральную эффективность, достигается при использовании алгоритма с подходящей диагональной матрицей регуляризации. Нгуен и Ле-Нгок [21] выводят формулу адаптивного РПФ (АРПФ) для системы с одноантенными приемниками (multi-user single-input multiple-output, MU-SIMO) с одноантенными пользователями. Алгоритм АРПФ эффективно использует разные параметры регуляризации для разных пользователей, учитывая сингулярные значения передаваемых слоёв (потоков). Мотивированные этим, мы исследуем префильтр АРПФ с диагональной регуляризацией, что соответствует общей теореме из [5] и обобщает результаты [21] на случай многоантенных приемников, в котором необходимо учитывать алгоритмы постфильтра[7] на стороне принимающих станций. Основная сложность такой постановки задачи заключается в том, что постфильтры реализуются каждой принимающей станцией из своих соображений (т.е. заведомо несогласованно) и более того не известны для передающей станции.

[6]ОСИШ отличается от обычного отношения сигнал-шум (ОСШ) тем, что часть помех, которые связаны с интерференцией и которые можно минимизировать на передатчике, вынесены в отдельное слагаемое: ОСИШ = сигнал (дБ) - (интерференция (дБ) + остальной шум (дБ).)

[7]Постфильтр – это обработка сигнала на стороне приемника, позволяющая отфильтровать шум и усилить сигнал в соответствии с используемой схемой модуляции и кодирования.

Результатом данной работы является адаптация формулы $\mathbf{W}_{ARZF}$ для многопользовательских многоантенных систем (multi-user multiple-input multiple-output, MU-MIMO) систем, а также подробное исследование свойств префильтра АРПФ, в том числе доказательство его асимптотической оптимальности в случае многоантенных приемников.

Статья организована следующим образом. В разделе 2.1 формулируется задача, включающая модель канала и системы, показатели качества и ограничения на мощность. Модель нисходящего канала с множеством антенн упрощается с помощью сингулярного разложения матрицы канала (раздел 2.1.1) и упрощенной модели постфильтра, асимптотически приближающей стандартные алгоритмы (раздел 2.1.3); показатели качества и ограничения по мощности обсуждаются в разделах 2.1.5 и 2.1.6. В разделе 2.2 определяются известные стандартные префильтры (СФ, ПФ, РПФ), в разделе 2.3 дается определение предлагаемому алгоритму АРПФ и исследуются его свойства и асимптотическая связь со стандартными алгоритмами. Сравнение алгоритмов на численных экспериментах в симуляторе "Quadriga" представлено в разделе 3.2. Заключение дано в разделе 4. Условные обозначения приведены в таблице 1.

В статье используются следующие обозначения. Рассматривается одна базовая станция с K пользователями, число передающих антенн — T, число приёмных антенн и передаваемых символов у пользователя k обозначаются как $R_k = 1, 2, 4$ и $L_k \leqslant R_k$, соответственно. Общее число приёмных антенн: $R = \sum_{k=1}^{K} R_k$, а число передаваемых символов: $L = \sum_{k=1}^{K} L_k$, при этом $L_k \leqslant R_k \leqslant T$. Жирными строчными буквами обозначаются векторы (строки или столбцы). Матрицы обозначаются жирными заглавными буквами и трактуются как совокупности векторов, например, матрица канала — это набор векторов-строк: $\mathbf{H} = [\mathbf{H}_1; \ldots; \mathbf{H}_R] \in \mathbb{C}^{R \times T}$, а матрица префильтра — набор векторов-столбцов: $\mathbf{W} = (\mathbf{W}_1, \ldots, \mathbf{W}_L) \in \mathbb{C}^{T \times L}$. Элементы матриц обозначаются обычными строчными буквами с двумя индексами: первый индекс — строка, второй — столбец: $\mathbf{H} = \{h_{rt}\}$, $\mathbf{W} = \{w_{tl}\}$, $r = 1, \ldots, R$, $t = 1, \ldots, T$, $l = 1, \ldots, L$. Эрмитово сопряжение обозначается как $\mathbf{H}^{\mathrm{H}} := \overline{\mathbf{H}}^{\mathrm{T}}$. Диагональные и блочно-диагональные матрицы записываются как $\mathbf{S}_k = \mathrm{diag}\{s_{k,1}, \ldots, s_{k,R_k}\}$ и $\mathbf{S} = \mathrm{bdiag}\{\mathbf{S}_1, \ldots, \mathbf{S}_K\}$, соответственно. Единичная матрица размера T: $\mathbf{I}_T = \mathrm{diag}\{1, \ldots, 1\} \in \mathbb{C}^{T \times T}$. След квадратной матрицы $\mathbf{A}$ обозначается как $tr\mathbf{A} = \sum_{k=1}^{K} a_{kk}$, а норма Фробениуса: $\|\mathbf{H}\| = \sqrt{\sum_{r=1,t=1}^{R,T} |h_{rt}|^2}$.

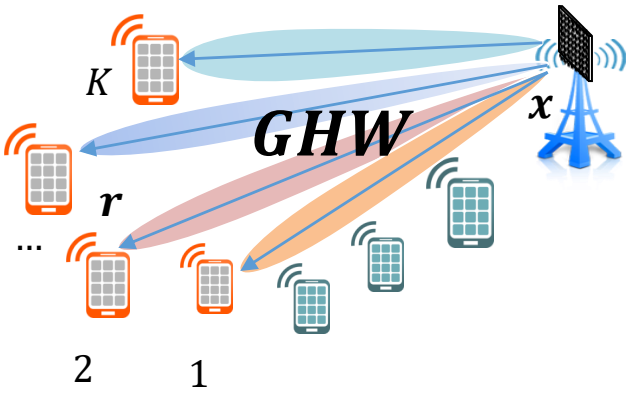

Рис. 1. Пример использования префильтра в системе с множеством антенн. Задача состоит в нахождении оптимальной матрицы префильтра $\mathbf{W}$ для системы при заданной функции спектральной эффективности и ограничениях (25).

Таблица 1. Обозначения и символы

Символ	Обозначение
$()^{\mathrm{H}}$	Эрмитово сопряжение (комплексное сопряжение + транспонирование)
$\mathbf{H}, \mathbf{W}$	Матрицы
$\mathbf{W}_n$	n-й столбец матрицы $\mathbf{W}$
$\mathbf{H}_m, \mathbf{W}^m$	m-я строка матриц $\mathbf{H}, \mathbf{W}$
h_{nm}, w_{nm}	Элемент на пересечении n-й строки и m-го столбца матриц $\mathbf{H}, \mathbf{W}$
$\mathbf{S} = \mathrm{diag}(s_1, ..., s_N)$	Диагональная матрица
K	Количество пользователей
T	Число передающих антенн
R	Общее число приёмных антенн
R_k	Число приёмных антенн у k-го пользователя
L	Общее число принимаемых символов в системе
L_k	Число принимаемых символов у k-го пользователя (ранг k-го пользователя)

2. Методы

2.1. Модель канала и системы

В соответствии с [2, 22, 12] рассматривается широковещательный канал с множеством антенн. Модель многопользовательской системы с множеством антенн описывается следующей линейной системой:

$$\mathbf{r} = \mathbf{G}(\mathbf{HWx} + \mathbf{n}) = \mathbf{GHWx} + \mathbf{Gn}, \tag{1}$$

где $\mathbf{x}, \mathbf{r} \in \mathbb{C}^L$ — это соответственно *векторы передаваемого и принимаемого сигнала*, $\mathbf{H} \in \mathbb{C}^{R \times T}$ — *матрица нисходящего канала*[8], $\mathbf{W} \in \mathbb{C}^{T \times L}$ — *матрица префильтра*, а $\mathbf{G} \in \mathbb{C}^{L \times R}$ — блочно-диагональная *матрица постфильтра*; *вектор шума* $\mathbf{n} \sim \mathcal{CN}(0, \sigma^2 \mathbf{I}_R)$ предполагается комплексно-нормальным с нулевым средним и дисперсией σ^2 (см. рис. 1). Линейный префильтр и постфильтр реализуются простыми матричными умножениями.

Постоянные T, R, L обозначают количество передающих антенн, общее количество приёмных антенн и количество передаваемых символов (количество потоков), соответственно. Обычно выполняется соотношение $L \leqslant R \leqslant T$. Каждая из матриц $\mathbf{G}, \mathbf{H}, \mathbf{W}$ раскладывается по K *пользователям*:

$$\mathbf{G} = \mathrm{bdiag}\{\mathbf{G}_1, \ldots, \mathbf{G}_K\}, \quad \mathbf{H} = [\mathbf{H}_1; \ldots; \mathbf{H}_K], \quad \mathbf{W} = (\mathbf{W}_1, \ldots, \mathbf{W}_K),$$

как показано на рис. 2, где $\mathbf{G}_k \in \mathbb{C}^{L_k \times R_k}$, $\mathbf{H}_k \in \mathbb{C}^{R_k \times T}$, $\mathbf{W}_k \in \mathbb{C}^{T \times L_k}$ — матрицы, соответствующие k-му пользователю.

2.1.1. Сингулярное разложение канала

Удобно [20] представлять матрицу канала пользователя k с помощью её сингулярного разложения:

$$\begin{aligned} \mathbf{H}_k = \mathbf{U}_k^{\mathrm{H}} \mathbf{S}_k \mathbf{V}_k, \quad \mathbf{U}_k \mathbf{U}_k^{\mathrm{H}} = \mathbf{U}_k^{\mathrm{H}} \mathbf{U}_k = \mathbf{I}_{R_k}, \\ \mathbf{S}_k = \mathrm{diag}\{s_1, \ldots, s_{R_k}\}, \quad \mathbf{V}_k \mathbf{V}_k^{\mathrm{H}} = \mathbf{I}_{R_k}. \end{aligned} \tag{2}$$

[8] Здесь и далее предполагается, что передающая сторона имеет идеальную информацию о состоянии всех нисходящих каналах. Это предположение оправдано для систем с разделением восходящей и нисходящей передач по времени, в которых можно использовать принцип взаимности и проводить непосредственные измерения соответствующих каналов. Каждый пользователь, в свою очередь, обладает только информацией о собственном канале с учетом примененного префильтра, но не знает информации о канале других пользователей.

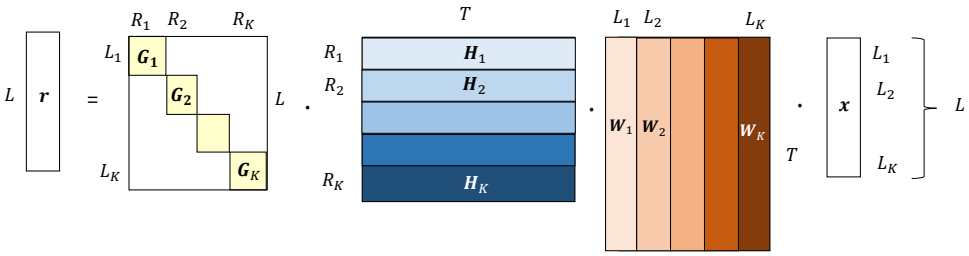

Рис. 2. Пример системы передачи в матричной форме. Многопользовательский префильтр $\mathbf{W}$ позволяет одновременно передавать различную информацию разным пользователям.

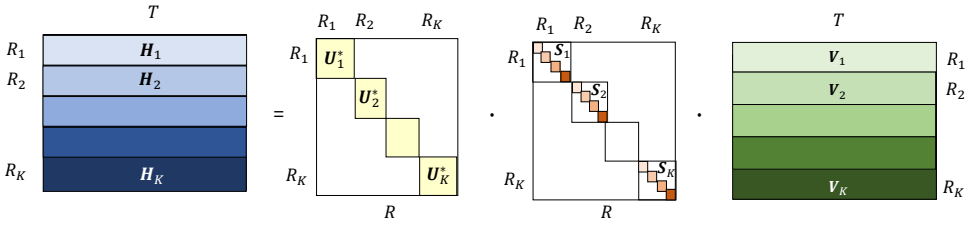

Рис. 3. Основное разложение матрицы канала.

Здесь *матрица канала* пользователя k, $\mathbf{H}_k \in \mathbb{C}^{R_k \times T}$, содержит по строкам векторы канала $\mathbf{H}_i \in \mathbb{C}^T$; сингулярные значения $\mathbf{S}_k \in \mathbb{C}^{R_k \times R_k}$ отсортированы по убыванию; $\mathbf{U}_k \in \mathbb{C}^{R_k \times R_k}$ — унитарная матрица левых сингулярных векторов; матрица $\mathbf{V}_k \in \mathbb{C}^{R_k \times T}$ состоит из *правых сингулярных векторов* — векторов-строк.

Объединяя все каналы пользователей, получаем следующее *разложение матрицы канала*: $\mathbf{H} = \mathbf{U}^{\mathrm{H}}\mathbf{S}\mathbf{V}$ (лемма 1 и рис. 3), где каждая из матриц разложения — $\mathbf{U}^{\mathrm{H}} \in \mathbb{C}^{R \times R}, \mathbf{S} \in \mathbb{C}^{R \times R}, \mathbf{V} \in \mathbb{C}^{R \times T}$ — состоит из K подматриц $\mathbf{U}_k^{\mathrm{H}} \in \mathbb{C}^{R_k \times R_k}, \mathbf{S}_k \in \mathbb{C}^{R_k \times R_k}, \mathbf{V}_k \in \mathbb{C}^{R_k \times T}$, содержащих векторы $\mathbf{U}_l^{\mathrm{H}} \in \mathbb{C}^{R_k}, s_l \in \mathbb{R}, \mathbf{V}_l \in \mathbb{C}^T$.

Лемма 1 (Основное разложение)**.** *Для линейной системы* $\mathbf{r} = \mathbf{G}(\mathbf{H}\mathbf{W}\mathbf{x} + \mathbf{n})$ *существует разложение матрицы канала в виде* $\mathbf{H} = \mathbf{U}^{\mathrm{H}}\mathbf{S}\mathbf{V}$,
где $\mathbf{S} = diag\{\mathbf{S}_1, \ldots, \mathbf{S}_K\} \in \mathbb{R}_+^{R \times R}$ *— диагональная матрица сингулярных значений,*
$\mathbf{V} = [\mathbf{V}_1, \ldots, \mathbf{V}_K] \in \mathbb{C}^{R \times T}$ *— конкатенация матриц сингулярных векторов для каналов отдельных пользователей,*
а $\mathbf{U}^{\mathrm{H}} = bdiag\{\mathbf{U}_1^{\mathrm{H}}, \ldots, \mathbf{U}_K^{\mathrm{H}}\} \in \mathbb{C}^{R \times R}$ *— блочно-диагональная унитарная матрица.*

Отметим, что это представление не является строгим сингулярным разложением всей матрицы $\mathbf{H}$, так как векторы $\mathbf{V}_{k,j}, \mathbf{V}_{l,i}$, соответствующие различным пользователям $k \neq l$, в общем случае не ортогональны. Тем не менее, данное представление обладает важными свойствами: матрица $\mathbf{S} = \mathrm{diag}(\mathbf{S}_k) \in \mathbb{C}^{R \times R}$ является *диагональной*, а $\mathbf{U} = \mathrm{bdiag}(\mathbf{U}_k) \in \mathbb{C}^{R \times R}$ — *блочно-диагональной унитарной матрицей*. Это позволяет компенсировать множитель $\mathbf{U}^{\mathrm{H}}\mathbf{S}$ на стороне приёмника (причём каждый пользователь обрабатывает только собственную часть $\mathbf{U}_k^{\mathrm{H}}\mathbf{S}_k$). Таким образом, на стороне передачи достаточно инвертировать только матрицу $\mathbf{V}$, которая значительно проще, чем $\mathbf{H}$: во-первых, её строки имеют единичную норму, а во-вторых, она является естественным объектом для задачи адаптации ранга [23]. Мы активно используем лемму 1 в дальнейшем.

2.1.2. Идея исследуемого алгоритма префильтра

Кратко сформулируем суть предлагаемого алгоритма. Интуитивная идея исследуемого метода префильтра заключается в том, чтобы обеспечить максимальную мощность сигнала при одновременном устранении так называемых межпользовательских помех. Известно, что максимальную мощность сигнала обеспечивает метод сопряжённого префильтра: $\mathbf{W}_{MRT}(\mathbf{V}) = \mathbf{V}^{\mathrm{H}}$, а полное устранение помех достигается с помощью псевдообратного префильтра: $\mathbf{W}_{ZF}(\mathbf{V}) = \mathbf{V}^{\mathrm{H}}(\mathbf{V}\mathbf{V}^{\mathrm{H}})^{-1}$ [3]. Однако у обоих методов имеются недостатки, которые могут быть устранены с помощью регуляризованного псевдообратного префильтра.

$$\mathbf{W}_{RZF}(\mathbf{V}) = \mathbf{V}^{\mathrm{H}}(\mathbf{V}\mathbf{V}^{\mathrm{H}} + \lambda\mathbf{I})^{-1},$$

где параметр регуляризации $\lambda > 0$ зависит от уровня шума и среднего значения потерь на пути сигнала [4].

Если у пользователей существенно различаются потери на пути, предпочтительнее использовать регуляризацию с диагональной матрицей. Нескалярную (матричную) регуляризацию предложил Э. Бьёрнсон в работе [5], где она была получена с помощью метода двойственности. Показано, что оптимальная регуляризация имеет диагональную форму (в общем случае с различными элементами). Однако полученный результат не носил конструктивного характера и не давал явной формулы или процедуры для построения оптимальной регуляризации.

В нашей работе мы исследуем явную эвристическую формулу для диагональной регуляризации — алгоритм адаптивного регуляризованного псевдообратного префильтра (АРПФ, для англоязычной литературы мы бы перевели как Adaptive Regularized Zero-Forcing, ARZF) для многопользовательской системы с множеством антенн.

$$\mathbf{W}_{ARZF} = \mathbf{V}^{\mathrm{H}}(\mathbf{V}\mathbf{V}^{\mathrm{H}} + \lambda\mathbf{S}^{-2})^{-1}.$$

Теоретическое обоснование формулы представлено далее в разделе 2.2. Численные эксперименты с использованием модели канала “Quadriga” [9] показывают, что рассматриваемый метод АРПФ превосходит известные скалярные алгоритмы РПФ (см. раздел 3.2).

2.1.3. Число передаваемых символов и сопряжённый постфильтр

В предыдущем разделе была введено понятие сингулярного разложения (см. уравнение 2). Обычно передатчик передаёт каждому пользователю несколько символов, это число (ранг пользователя) меньше числа приёмных антенн пользователя ($L_k \leqslant R_k$). В таком случае естественно выбирать для передачи первые L_k векторов из матрицы $\mathbf{V}_k$, соответствующих L_k наибольшим сингулярным числам из матрицы $\mathbf{S}_k$.

Обозначим через $\widetilde{\mathbf{S}}_k \in \mathbb{C}^{L_k \times L_k}$ L_k наибольших сингулярных чисел из $\mathbf{S}_k$, через $\widetilde{\mathbf{U}}_k^{\mathrm{H}} \in \mathbb{C}^{R_k \times L_k}$ и $\widetilde{\mathbf{V}}_k \in \mathbb{C}^{L_k \times T}$ — соответствующие им первые левые и правые сингулярные векторы:

$$\begin{aligned} \widetilde{\mathbf{S}}_k &= \mathrm{diag}\{s_{k,1}, \ldots, s_{k,L_k}\}, \\ \widetilde{\mathbf{U}}_k^{\mathrm{H}} &= (\mathbf{u}_{k,1}^{\mathrm{H}}, \ldots, \mathbf{u}_{k,L_k}^{\mathrm{H}}), \\ \widetilde{\mathbf{V}}_k &= [\mathbf{v}_{k,1}; \ldots; \mathbf{v}_{k,L_k}], \end{aligned} \tag{3}$$

где волнистая черта $\widetilde{\cdot}$ означает сокращённое представление по сингулярным векторам, то есть $rank(\widetilde{\mathbf{V}}_k) = L_k \leqslant R_k = rank(\mathbf{V}_k)$.

Числа L_k (и, соответственно, выбор $\widetilde{\mathbf{V}}_k$) определяются в задаче выбора ранга, которая вместе с планировщиком решается до этапа выбора префильтра. За дополнительной информацией по адаптации ранга см., например, [23]. Далее считаем, что L_k и $\widetilde{\mathbf{V}}_k$ уже выбраны.

После выбора префильтра и передачи на стороне пользователя k необходимо выбрать матрицу постфильтра $\mathbf{G}_k \in \mathbb{C}^{L_k \times R_k}$, которая учитывает ранг пользователя L_k. Способ, которым пользователь выполняет процедуру постфильтра, существенно влияет на итоговую производительность системы, и разные алгоритмы постфильтра требуют различных оптимальных префильтров (см. [4], где префильтр является функцией матрицы постфильтра). Оптимально было бы согласованно выбирать префильтр и постфильтр, но это затруднительно из-за распределённого характера беспроводной связи. Однако существуют идеи по настройке префильтра на передатчике при предполагаемом способе постфильтра на приёмной стороне [10]. В данной работе мы не рассматриваем такой подход, хотя он может быть использован для дальнейшего улучшения.

Для проведения аналитических выкладок мы будем предполагать использование *сопряжённого постфильтра* (который достаточно хорошо

приближает реальные алгоритмы и при этом сильно упрощает задачу, см. [24]) следующего вида:

$$\mathbf{G}_k^C = \widetilde{\mathbf{S}}_k^{-1}\widetilde{\mathbf{U}}_k \in \mathbb{C}^{L_k \times R_k} \Longleftrightarrow \mathbf{G}^C := \widetilde{\mathbf{S}}^{-1}\widetilde{\mathbf{U}} \in \mathbb{C}^{L \times R}, \tag{4}$$

где матрица $\widetilde{\mathbf{U}}_k \in \mathbb{C}^{L_k \times R_k}$ содержит первые L_k сингулярных векторов, а диагональная матрица $\widetilde{\mathbf{S}}_k \in \mathbb{C}^{L_k \times L_k}$ — соответствующие сингулярные числа. Блочно-диагональная матрица $\widetilde{\mathbf{U}}$ составлена из блоков $\widetilde{\mathbf{U}}_k$, а диагональная матрица $\widetilde{\mathbf{S}}^{-1}$ из блоков $\widetilde{\mathbf{S}}_k^{-1}$. Таким образом, матрица постфильтра $\mathbf{G}^C$ также блочно-диагональна: $\mathbf{G}^C := \widetilde{\mathbf{S}}^{-1}\widetilde{\mathbf{U}} \in \mathbb{C}^{L \times R}$.

Теорема 1. *Сопряжённый постфильтр удаляет неиспользуемые сингулярные векторы:* $\mathbf{G}^C\mathbf{H} = \widetilde{\mathbf{V}}$*, и уравнение модели* (1) *принимает вид:*

$$\mathbf{r} = \widetilde{\mathbf{V}}\mathbf{W}\mathbf{x} + \tilde{\mathbf{n}}, \quad \tilde{\mathbf{n}} := \widetilde{\mathbf{S}}^{-1}\widetilde{\mathbf{U}}\mathbf{n}. \tag{5}$$

Доказательство. Используя лемму (1), получаем:

$$\mathbf{G}_k^C\mathbf{H}_k = \widetilde{\mathbf{S}}_k^{-1}\widetilde{\mathbf{U}}_k\mathbf{U}_k^{\mathrm{H}}\mathbf{S}_k\mathbf{V}_k = \widetilde{\mathbf{S}}_k^{-1}\begin{bmatrix}\mathbf{I}_{L_k} & \mathbf{0}\end{bmatrix}\mathbf{S}_k\mathbf{V}_k = \widetilde{\mathbf{S}}_k^{-1}\widetilde{\mathbf{S}}_k\widetilde{\mathbf{V}}_k = \widetilde{\mathbf{V}}_k, \tag{6}$$

что при объединении по всем пользователям $k = 1, \ldots, K$ даёт равенство (5). □

Следствие 1. *При предположении о гауссовом независимом шуме* $\mathbf{n} \sim \mathcal{CN}(0, \sigma^2\mathbf{I}_R)$*, распределение эффективного шума* $\tilde{\mathbf{n}} \in \mathbb{C}^{L \times 1}$ *(который появляется при сопряжённом постфильтре) описывается как:* $\tilde{\mathbf{n}} \sim \mathcal{CN}(0, \sigma^2\widetilde{\mathbf{S}}^{-2})$.

Доказательство.

$$\mathbb{E}[\tilde{\mathbf{n}}\tilde{\mathbf{n}}^{\mathrm{H}}] = \mathbb{E}[\widetilde{\mathbf{S}}^{-1}\widetilde{\mathbf{U}}\mathbf{n}\mathbf{n}^{\mathrm{H}}\widetilde{\mathbf{U}}^{\mathrm{H}}\widetilde{\mathbf{S}}^{-1}] = \widetilde{\mathbf{S}}^{-1}\widetilde{\mathbf{U}}\mathbb{E}[\mathbf{n}\mathbf{n}^{\mathrm{H}}]\widetilde{\mathbf{U}}^{\mathrm{H}}\widetilde{\mathbf{S}}^{-1}$$
$$= \sigma^2\widetilde{\mathbf{S}}^{-1}\mathbf{I}_L\widetilde{\mathbf{S}}^{-1} = \sigma^2\widetilde{\mathbf{S}}^{-2}. \tag{7}$$

□

Замечание 1. *Следствие 1 играет ключевую роль в нашей работе, поскольку оно даёт основание использовать матрицу* $\widetilde{\mathbf{S}}^{-2} \in \mathbb{C}^{L \times L}$ *в регуляризационном слагаемом префильтра с целью корректного учёта эффективного шума* $\widetilde{\mathbf{S}}^{-1}\widetilde{\mathbf{U}}\mathbf{n}$.

Замечание 2. *Формулируемая теорема существенно упрощает исходную задачу, уменьшает её размерность и позволяет унифицировать обозначения. В частности, мы можем работать с рангом пользователей размерности* L_k *и* L*, вместо пространства антенн пользователя.*

Заметим также, что достаточно выполнять лишь частичное сингулярное разложение канала $\mathbf{H}_k \in \mathbb{C}^{R_k \times T}$*, сохраняя только первые* L_k *сингулярных значений и векторов:*

$$\mathbf{H}_k \approx \widetilde{\mathbf{U}}_k^{\mathrm{H}} \widetilde{\mathbf{S}}_k \widetilde{\mathbf{V}}_k. \tag{8}$$

В связи с этим далее по тексту мы будем опускать волнистую черту и обозначать $\mathbf{U}_k, \mathbf{S}_k, \mathbf{V}_k$ *вместо* $\widetilde{\mathbf{U}}_k, \widetilde{\mathbf{S}}_k, \widetilde{\mathbf{V}}_k$ *соответственно.*

Замечание 3. *Введённый сопряжённый постфильтр является "идеальным" и не может быть реализован на практике. Однако можно показать, что реалистичные алгоритмы постфильтра, такие как постфильтр среднеквадратичной ошибки (Minimum Mean Square Error, MMSE) или постфильтр с подавлением помех (Interference Rejection Combining, IRC) [25] , зачастую демонстрируют схожее поведение. В численных экспериментах для сравнения различных префильтров используется постфильтр среднеквадратичной ошибки.*

2.1.4. Постфильтр для минимизации среднеквадратичной ошибки

Целью процедуры постфильтра сигнала является псевдообращение произведения матриц канала и префильтра. Наиболее распространённой формой постфильтра $\mathbf{G}$ является матрица минимизации среднеквадратичной ошибки [26]. В статистике и обработке сигналов постфильтр среднеквадратичной ошибки — это метод оценки, минимизирующий функцию среднеквадратичной ошибки, которая, в свою очередь, служит общей мерой качества оценки предсказанных значений зависимой переменной.

Определение постфильтра среднеквадратичной ошибки реализуется следующим образом:

$$\mathbf{G}_k^{MMSE}(\mathbf{A}_k) = \mathbf{A}_k^{\mathrm{H}} \left(\mathbf{A}_k \mathbf{A}_k^{\mathrm{H}} + \sigma^2 \mathbf{I} \right)^{-1}, \quad \mathbf{A}_k = \mathbf{H}_k \mathbf{W}_k. \tag{9}$$

Параметр P — мощность базовой станции, а σ^2 — мощность шума в системе. Метод постфильтра среднеквадратичной ошибки предполагает устранение шума при допущении его одинаковости для всех символов: $\mathbf{n} \sim \mathcal{CN}(0, \sigma^2 \mathbf{I}_R)$, что может нарушаться на практике.

Лемма 2. *Для системы* (1)*:* $\mathbf{r}_k = \mathbf{G}_k \mathbf{H}_k \mathbf{W} \mathbf{x} + \mathbf{G}_k \mathbf{n}_k \in \mathbb{C}^{L_k}$ *с распределением шума* $\mathbf{n} \sim \mathcal{CN}(0, \sigma^2 \mathbf{I}_R)$ *и префильтром* $\mathbf{W} = \mathbf{H}^{+} = \mathbf{H}^{\mathrm{H}} (\mathbf{H}\mathbf{H}^{\mathrm{H}})^{-1}$*, матрица* (9)*:* $\mathbf{G}^{MMSE}$ *минимизирует норму в квадрате:* $\mathbb{E}_{\mathbf{n},\mathbf{x}} \|\mathbf{r}_k - \mathbf{x}\|^2$*,* $k = 1 \dots K$*.*

Доказательство. Подставим выражение для системы $\mathbf{r}_k = \mathbf{G}_k \mathbf{H}_k \mathbf{W} \mathbf{x} + \mathbf{G}_k \mathbf{n}_k$ в функцию потерь:

$$\mathbb{E}_{\mathbf{n} \sim \mathcal{CN}(0, \sigma^2 \mathbf{I}_R)} \|\mathbf{r}_k - \mathbf{x}\|^2 = \mathbb{E}_{\mathbf{n} \sim \mathcal{CN}(0, \sigma^2 \mathbf{I}_R)} \|\mathbf{G}_k \mathbf{H}_k \mathbf{W} \mathbf{x} - \mathbf{x} + \mathbf{G}_k \mathbf{n}_k\|^2.$$

Второе слагаемое обнуляется благодаря математическому ожиданию шума с нулевым средним. Третее слагаемое также исчезает.

Раскроем скобки, используя формулу квадрата нормы суммы:

$$\mathbb{E}_{\mathbf{n}\sim\mathcal{CN}(0,\sigma^2\mathbf{I}_R)}\|(\mathbf{G}_k\mathbf{H}_k\mathbf{W}-\mathbf{I})\mathbf{x}+\mathbf{G}_k\mathbf{n}_k\|^2 = \|(\mathbf{G}_k\mathbf{H}_k\mathbf{W}-\mathbf{I})\mathbf{x}\|^2 + \\ + 2\mathbb{E}_{\mathbf{n}\sim\mathcal{CN}(0,\sigma^2\mathbf{I}_R)}\Re\{\langle\mathbf{G}_k\mathbf{H}_k\mathbf{W}\mathbf{x}-\mathbf{x},\mathbf{G}_k\mathbf{n}_k\rangle\} + \mathbb{E}_{\mathbf{n}\sim\mathcal{CN}(0,\sigma^2\mathbf{I}_R)}\|\mathbf{G}_k\mathbf{n}_k\|^2 = \\ = \|(\mathbf{G}_k\mathbf{H}_k\mathbf{W}-\mathbf{I})\mathbf{x}\|^2+\sigma^2\|\mathbf{G}_k\|^2. \quad (10)$$

Теперь возьмём математическое ожидание по символам $\mathbf{x}$ с условием $\mathbb{E}_{\mathbf{x}\sim\mathcal{CN}(0,\mathbf{I}_L)}\mathbf{x}\mathbf{x}^{\mathrm{H}}=\mathbf{I}$:

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{CN}(0,\mathbf{I}_L)}\|(\mathbf{G}_k\mathbf{H}_k\mathbf{W}-\mathbf{I})\mathbf{x}\|^2+\sigma^2\|\mathbf{G}_k\|^2 = \\ = \|\mathbf{G}_k\mathbf{H}_k\mathbf{W}-\mathbf{I}\|^2+\sigma^2\|\mathbf{G}_k\|^2 \to \min_{\mathbf{G}_k}. \quad (11)$$

Если выполнены условия: $\mathbf{H}_k\mathbf{W}=\mathbf{H}_k\mathbf{W}_k$, где $\mathbf{W}=\mathbf{H}^{\mathrm{H}}(\mathbf{H}\mathbf{H}^{\mathrm{H}})^{-1}$, то функция принимает вид:

$$\|\mathbf{G}_k\mathbf{H}_k\mathbf{W}_k-\mathbf{I}\|^2+\sigma^2\|\mathbf{G}_k\|^2 \to \min_{\mathbf{G}_k} \quad (12)$$

Вычислим градиент функции (12) и приравняем его к нулю:

$$\nabla_{\mathbf{G}_k}\{\|\mathbf{G}_k\mathbf{H}_k\mathbf{W}_k-\mathbf{I}\|^2+\sigma^2\|\mathbf{G}_k\|^2\} = \\ 2(\mathbf{G}_k\mathbf{H}_k\mathbf{W}_k-\mathbf{I})(\mathbf{H}_k\mathbf{W}_k)^{\mathrm{H}}+2\sigma^2\mathbf{G}_k = \\ = 2\mathbf{G}_k(\mathbf{H}_k\mathbf{W}_k)(\mathbf{H}_k\mathbf{W}_k)^{\mathrm{H}}-2(\mathbf{H}_k\mathbf{W}_k)^{\mathrm{H}}+2\sigma^2\mathbf{G}_k=0 \quad (13)$$

Переносим слагаемые с $\mathbf{G}_k$ в левую часть:

$$\mathbf{G}_k((\mathbf{H}_k\mathbf{W}_k)(\mathbf{H}_k\mathbf{W}_k)^{\mathrm{H}}+\sigma^2\mathbf{I}) = (\mathbf{H}_k\mathbf{W}_k)^{\mathrm{H}} \quad (14)$$

В итоге выражаем из уравнения $\mathbf{G}_k$:

$$\widehat{\mathbf{G}}_k^{MMSE} = \mathbf{G}_k = (\mathbf{H}_k\mathbf{W}_k)^{\mathrm{H}}((\mathbf{H}_k\mathbf{W}_k)(\mathbf{H}_k\mathbf{W}_k)^{\mathrm{H}}+\sigma^2\mathbf{I})^{-1} \quad (15)$$

Получаем желаемое решение (9). □

2.1.5. Оценки качества

Для оценки качества методов префильтра используются следующие функционалы. Эти функции зависят не от конкретных передаваемых символов $\mathbf{x}\in\mathbb{C}^{L\times 1}$, а от их распределения [5]. Таким образом, получаем общую

характеристику для всех возможных символов, которые могут быть переданы с использованием данной матрицы префильтра.

Рассмотрим сквозную нумерацию символов $l = 1, \ldots, L$ для всех пользователей, и определим индексную функцию $k = k(l)$, которая возвращает номер пользователя, получающего символ l. Функционал отношения сигнал-интерференция-шум (ОСИШ) для l-го символа пользователя $k = k(l)$ определяется как:

$$SINR_l(\mathbf{W}, \mathbf{H}_k, \mathbf{G}_l, \sigma^2) := \frac{|\mathbf{G}_l \mathbf{H}_k \mathbf{W}_l|^2}{\sum_{i \neq l}^{L} |\mathbf{G}_l \mathbf{H}_k \mathbf{W}_i|^2 + \sigma^2 \|\mathbf{G}_l\|^2}. \tag{16}$$

Формула (16) показывает соотношение между полезной частью сигнала и помехами. Она зависит от всей матрицы префильтра $\mathbf{W} \in \mathbb{C}^{T \times L}$, где вектор $\mathbf{W}_l \in \mathbb{C}^{T \times 1}$ соответствует префильтру для l-го символа, матрицы канала $\mathbf{H}_k \in \mathbb{C}^{R_k \times T}$ пользователя k, вектора постфильтра $\mathbf{G}_l \in \mathbb{C}^{1 \times R_k}$, и уровня шума после постфильтра: $\mathbb{E}[\mathbf{G}_l \mathbf{n}] = \sigma^2 \|\mathbf{G}_l\|^2$. Формула (16) может быть эффективно вычислена для всех L слоёв при помощи матричных операций и суммирования.

Формулу можно упростить, используя теорему 1:

Следствие 2. *Для сингулярного разложения (2) и предположения сопряжённого постфильтра (4), ОСИШ выражается как:*

$$SINR_l^C(\mathbf{W}, \mathbf{V}_l, s_l, \sigma^2) = SINR_l(\mathbf{W}, \mathbf{H}_k, \mathbf{G}_l^C, \sigma^2) = \frac{|\mathbf{V}_l \mathbf{W}_l|^2}{\sum_{i \neq l}^{L} |\mathbf{V}_l \mathbf{W}_i|^2 + \sigma^2 / s_l^2}. \tag{17}$$

Важным критерием производительности сети является *спектральная эффективность* (Specrtal Efficiency, SE) пользователя k, отражающая максимальную скорость передачи информации на заданной полосе частот. При передаче одного символа ($L_k = 1$), она ограничена энтропией Шеннона и выражается через ОСИШ следующим образом:

$$SE(SINR) := \log_2(1 + SINR). \tag{18}$$

Это является теоретическим пределом передаваемой информации. Современные схемы модуляции и кодирования, в сочетании с механизмами гибридного автоматического запроса повторной передачи (Hybrid Automatic Repeat Request, HARQ) и управлением блочной ошибки (Block Error Rate, BLER), позволяют достичь скоростей, близких к этому пределу. Обратите внимание, что ОСИШ здесь используется в линейной шкале, а не в децибелах (дБ).

При передаче нескольких символов спектральная эффективность пользователя k в общем случае не является суммой спектральных эффективностей разных символов, так как используется общий транспортный

блок, к которому применяются общие алгоритмы кодирования и модуляции. Обычно вводится *эффективный* ОСИШ как функция от ОСИШ по символам:

$$SINR_k^{eff} = f(SINR_1, \ldots, SINR_{L_k}),$$

и тогда по формуле (18) получаем:

$$SE_k(\mathbf{W}, \mathbf{H}_k, \mathbf{G}_k, \sigma^2) = L_k \, \mathcal{S}\Big(SINR_k^{eff}(\mathbf{W}, \mathbf{H}_k, \mathbf{G}_k, \sigma^2)\Big). \tag{19}$$

Существует множество подходов к оценке эффективного ОСИШ для разных схем квадратурно-амплитудной модуляции (КАМ) КАМ64 и КАМ256 (см. [27]); в численных экспериментах мы используем модель КАМ256.

Также рассмотрим приближённую формулу с использованием геометрического среднего по ОСИШ символов. Это эквивалентно обычному среднему по символам в дБ. Данное эвристическое приближение будет использоваться в градиентной оптимизации:

$$SINR_k^{eff}(\mathbf{W}, \mathbf{H}_k, \mathbf{G}_k, \sigma^2) \approx \Big(\prod\nolimits_{l \in \mathcal{L}_k} SINR_l(\mathbf{W}, \mathbf{H}_k, \mathbf{G}_l, \sigma^2)\Big)^{\frac{1}{L_k}}. \tag{20}$$

Наиболее общий вид задачи — многокритериальная оптимизация вектора $(SE_1, ..., SE_K)$. Для такой постановки может быть проведён анализ по Парето (см. [28, 29]), который, однако, не даёт уникального решения. Поэтому часто задача сводится к однокритериальной оптимизации: $J = J(SE_1, ..., SE_K) \to \max$ или $J = J(SINR_1, ..., SINR_K)$ [5]. Один из возможных вариантов — максимизация суммы спектральных эффективностей:

$$J^{SE}(\mathbf{W}) := SE(\mathbf{W}, \mathbf{H}, \mathbf{G}, \sigma^2) = \sum_{k=1}^{K} SE_k(\mathbf{W}, \mathbf{H}_k, \mathbf{G}_k, \sigma^2). \tag{21}$$

Такой критерий является естественным, поскольку скорости передачи данных по пользователям являются аддитивными. Возможны и другие цели, например, оптимизация качества самых плохих пользователей. Подробный анализ таких постановок приведён в [12, разд. 7], включая:

$$SE_{\min} = \min_k SE_k \to \max \quad \text{или} \quad SINR_{\min} = \min_{1 \leqslant j \leqslant L} SINR_j \to \max. \tag{22}$$

Наконец, введём *однопользовательский* ОСИШ (Single User SINR, SUSINR) для пользователя k (эти метрики будут использоваться для оценки результатов на тестах).

$$SUSINR_k(\mathbf{S}_k, \sigma^2, P) := \frac{P}{L_k \sigma^2} \left(\prod\nolimits_{l \in \mathcal{L}_k} s_l^2 \right)^{\frac{1}{L_k}}. \tag{23}$$

2.1.6. Постановка задачи и ограничения по мощности

Прежде всего, предполагается, что полная матрица канала $\mathbf{H}$, число пользователей K и их ранги L_k — известные заранее величины. Это означает, что задачи планировщика — каких пользователей обслуживать, и выбора ранга — какой ранг назначается каждому пользователю — уже решены. В реальных сетях такая последовательность шагов стандартна. Проблемы планировщика и выбора ранга представляют собой сложные задачи управления радиоресурсами и выходят за рамки данного исследования (см., например, [30] и ссылки внутри для задачи планировщика, [23], [10] — для задачи выбора ранга).

Эти алгоритмы влияют на свойства матрицы $\mathbf{H}$, например, планировщик может выбирать только пользователей с достаточно малыми корреляциями: $\|\mathbf{C}\| = \|\mathbf{V}\mathbf{V}^{\mathrm{H}} - \mathbf{I}_R\| \leqslant \varepsilon$. Мы учитываем это и рассматриваем сценарии с малой корреляцией каналов пользователей.

Далее, используем модель канала в виде (1), что, в частности, означает точные измерения канала. Для упрощения задачи предполагаем, что стратегия постфильтра $\mathbf{G} = \mathbf{G}(\mathbf{H}, \mathbf{W})$ — известная функция, причём применение сопряжённого постфильтра (4) упрощает модель канала до вида (5). На основе этой модели вычисляется *ОСИШ* передаваемых символов по формуле (17) и эффективный *ОСИШ* пользователей по приближённой формуле (20).

Обозначим полную мощность системы через P и предполагаем, что вектор передаваемых символов нормирован: $\mathbb{E}[\mathbf{x}\mathbf{x}^{\mathrm{H}}] = \mathbf{I}_L$. Тогда ограничения по мощности на префильтр могут быть следующими:

$$\|\mathbf{W}\|^2 \leqslant P, \qquad \text{или} \qquad \|(w_{t1}, \ldots, w_{tL})\|^2 \leqslant P/T, \quad t = 1, ..., T, \tag{24}$$

что соответствует, соответственно, полному ограничению по мощности и более реалистичному ограничению по мощности на каждую антенну [12].

Цель состоит в нахождении матрицы префильтра, максимизирующей суммарную спектральную эффективность (19) **при выполнении ограничений по мощности** (24), то есть:

$$\begin{aligned} J^{SE}(\mathbf{W}) &:= SE^C(\mathbf{W}), \\ \mathbf{W} &= \operatorname{argmax}_{\mathbf{W}} J^{SE}(\mathbf{W}), \\ \text{при условии:} &\|\mathbf{W}\|^2 \leqslant P, \end{aligned} \tag{25}$$

Даже после всех вышеуказанных упрощений, сформулированная задача остаётся слишком сложной для аналитического решения. Кроме того, она не является выпуклой или вогнутой, поэтому может иметь множество существенно различных локальных максимумов. Поэтому стратегия

данного исследования заключается в изучении эвристической формулы, превосходящей известные алгоритмы по результатам достоверного моделирования.

После задания конкретной формы префильтра $\mathbf{W} := \mu\mathbf{W}$ можно всегда выполнить ограничение по мощности, нормировкой на коэффициент μ, например, для случая (24):

$$\mu = \frac{\sqrt{P}}{\|\mathbf{W}\|} \quad \text{или} \quad \mu = \frac{\sqrt{P/T}}{\max\limits_{t=1,\dots,T}\{\|(w_{t1},\dots,w_{tL})\|\}}. \tag{26}$$

В моделировании мы используем более реалистичное ограничение по мощности на антенну.

Ниже сформулирован эвристический алгоритм, идея которого мотивирована упрощениями, приведёнными в следствии 1 и замечании 1. Теоретическим обоснованием служит модельная задача минимизации средней квадратичной ошибки, см., например, [4]:

$$\begin{aligned} J^{MSE}(\mathbf{W}) &:= \mathbb{E}_{\mathbf{x},\mathbf{n}}[\|\mathbf{r}(\mathbf{W}) - \mathbf{x}\|^2], \\ \mathbf{W} &= \operatorname{argmin}_{\mathbf{W}} J^{MSE}(\mathbf{W}), \\ \text{при условии:} &\|\mathbf{W}\|^2 \leqslant P, \end{aligned} \tag{27}$$

где $\mathbf{r}(\mathbf{W})$ определяется моделью канала (1) или (5).

2.2. Базовые префильтры

В данном разделе приведены известные эталонные алгоритмы префильтра, а также представлено предлагаемое решение.

2.2.1. Сопряжённый префильтр

Алгоритм сопряжённого префильтра (Maximum Ratio Transmission, MRT) использует веса одного пользователя, взятые как сопряжённо-транспонированная матрица $\mathbf{V}^{\mathrm{H}}$ из сингулярного разложения. Такой подход приводит к максимизации мощности полезного сигнала одного пользователя, полностью игнорируя межпользовательские помехи. Метод сопряжённого префильтра особенно эффективен в условиях, когда уровень шума превышает уровень межпользовательских помех [3]:

$$\mathbf{W}_{MRT}(\mathbf{V}) = \mu\mathbf{V}^{\mathrm{H}}, \tag{28}$$

где нормировочный множитель μ определяется из условия ограничения мощности (26).

Этот алгоритм приводит к формированию взаимно мешающих каналов, исходя из модели (5):

$$\mathbf{r} = \mathbf{V}\mathbf{W}\mathbf{x} + \mathbf{S}^{-1}\mathbf{U}\mathbf{n} = \mu\mathbf{V}\mathbf{V}^{\mathrm{H}}\mathbf{x} + \mathbf{S}^{-1}\mathbf{U}\mathbf{n}.$$

2.2.2. Псевдообратный префильтр

Следующая модификация алгоритма префильтра выполняет декорреляцию передаваемых символов с помощью обратной корреляционной матрицы векторов канала. Такая конструкция префильтра направляет лучи сигнала к пользователям, не создавая между ними взаимных помех. В отличие от метода сопряжённого префильтра, метод зануления помех псевдообратного префильтра (Zero-Forcing, ZF) предпочтителен в условиях, когда межпользовательские помехи превышают уровень шума. В этом случае качество спектральной эффективности улучшается за счёт устранения этих помех [3]:

$$\mathbf{W}_{ZF}(\mathbf{V}) = \mu\mathbf{V}^{\dagger} = \mu\mathbf{V}^{\mathrm{H}}(\mathbf{V}\mathbf{V}^{\mathrm{H}})^{-1}. \tag{29}$$

Соответствующая модель приёмника будет следующей:

$$\mathbf{r} = \mathbf{V}\mathbf{W}\mathbf{x} + \mathbf{S}^{-1}\mathbf{U}\mathbf{n} = \mu\mathbf{V}\mathbf{V}^{\mathrm{H}}(\mathbf{V}\mathbf{V}^{\mathrm{H}})^{-1}\mathbf{x} + \mathbf{S}^{-1}\mathbf{U}\mathbf{n} = \mu\mathbf{x} + \mathbf{S}^{-1}\mathbf{U}\mathbf{n}.$$

Обозначим $\mathbf{F} = \mathbf{U}\mathbf{H} = \mathbf{S}\mathbf{V}$.

Теорема 2. *Предположим, что матрица канала имеет вид* $\mathbf{H} = \mathbf{U}^{\mathrm{H}}\mathbf{S}\mathbf{V}$ *(см. Лемму 1). Тогда для метода зануления помех выполняется следующее равенство:*

$$\mathbf{W}_{ZF}(\mathbf{F})\mathbf{S} = \mathbf{W}_{ZF}(\mathbf{V}). \tag{30}$$

Доказательство.

$$\begin{aligned}\mathbf{W}_{ZF}(\mathbf{F})\mathbf{S} &= \mathbf{F}^{\mathrm{H}}(\mathbf{F}\mathbf{F}^{\mathrm{H}})^{-1}\mathbf{S} = \mathbf{V}^{\mathrm{H}}\mathbf{S}(\mathbf{S}\mathbf{V}\mathbf{V}^{\mathrm{H}}\mathbf{S})^{-1}\mathbf{S} = \\ &= \mathbf{V}^{\mathrm{H}}\mathbf{S}\mathbf{S}^{-1}(\mathbf{V}\mathbf{V}^{\mathrm{H}})^{-1}\mathbf{S}^{-1}\mathbf{S} = \mathbf{V}^{\mathrm{H}}(\mathbf{V}\mathbf{V}^{\mathrm{H}})^{-1} = \mathbf{W}_{ZF}(\mathbf{V}). \end{aligned} \tag{31}$$

□

2.2.3. Регуляризованный псевдообратный фильтр

В геометрическом смысле в методе ПФ (29) лучи направляются не строго на пользователей, а с отклонением, что уменьшает полезный сигнал. Следующая модификация корректирует направление лучей, допуская некоторое межпользовательское вмешательство, что значительно увеличивает полезную нагрузку.

В практическом смысле, в методе ПФ может не существовать правой обратной матрицы канала или матрица $\mathbf{V}\mathbf{V}^{\mathrm{H}}$ может быть плохо обусловленной, что ухудшает работу метода ПФ. Существует множество практических решений этой проблемы, основанных на регуляризации:

$$\mathbf{W}_{RZF}(\mathbf{V}) = \mu\mathbf{V}^{\mathrm{H}}(\mathbf{V}\mathbf{V}^{\mathrm{H}} + \lambda\mathbf{I})^{-1}. \tag{32}$$

Метод регуляризованного псевдообратного префильтра (Regularized Zero-Forcing, RZF) является самым распространённым в практических системах, и именно его мы используем в качестве основного эталонного метода. В качестве базовой настройки мы берём аналитическую форму регуляризации: $\lambda = \frac{L\sigma^2}{P}$ [8].

Этот метод не устраняет полностью межпользовательские и многослойные помехи, а допускает их в разумных пределах для увеличения мощности полезного сигнала. Он представляет собой компромисс между методами сопряжённого префильтра и псевдообратного префильтра [5], балансируя между максимизацией сигнала и минимизацией интерференции, поэтому параметр регуляризации должен быть выбран в зависимости от уровня шума.

Метод РПФ обладает следующими предельными свойствами [3]: при $\sigma^2 \to \infty$ он переходит в $\mathbf{W}_{MRT} = \mu\mathbf{V}^{\mathrm{H}}$ — оптимальный в условиях *низкого ОСИШ*. При $\sigma^2 = 0$ формула совпадает с псевдообратным префильтром: $\mathbf{W}_{ZF} = \mu\mathbf{V}^{\mathrm{H}}(\mathbf{V}\mathbf{V}^{\mathrm{H}})^{-1}$, которое оптимально при *высоком ОСИШ*.

Префильтр на основе ненормализованной матрицы канала [6], в случае когда количество передаваемых символов L меньше количества антенн приёмника, можно записать в следующем виде:

$$\mathbf{W}_{RZF}(\mathbf{F}) = \mu\mathbf{F}^{\mathrm{H}}(\mathbf{F}\mathbf{F}^{\mathrm{H}} + \lambda\mathbf{I})^{-1}, \tag{33}$$

где параметр $\mathbf{F} = \mathbf{S}\mathbf{V}$, а $\lambda = \frac{L\sigma^2}{P}$ [8] учитывает уровень шума $E[\mathbf{n}\mathbf{n}^{\mathrm{H}}] = \sigma^2\mathbf{I}_R$. На практике это эквивалентно использованию ненормализованной матрицы канала $\mathbf{H}$, которая в нашей модели заменяется матрицей $\mathbf{F}$ после разбиения канала пользователя на несколько потоков.

Сформулируем следующий известный факт о методе РПФ (32).

Теорема 3. *Рассмотрим разложение канала* $\mathbf{H} = \mathbf{U}^{\mathrm{H}}\mathbf{S}\mathbf{V}$ *из Леммы 1. Префильтр* $\mathbf{W}_{RZF}(\mathbf{V})$ *с любым параметром* $\lambda > 0$ *является решением следующей задачи оптимизации:*

$$\mathbf{W}_{RZF}(\mathbf{V}) = argmin_{\mathbf{W}} J(\mathbf{W}), \qquad J(\mathbf{W}) = \|\mathbf{V}\mathbf{W} - \mathbf{I}\|_2^2 + \lambda\|\mathbf{W}\|_2^2. \tag{34}$$

Доказательство. Вычисляя градиент и приравнивая его к нулю, получаем:

$$\nabla J(\mathbf{W}) = 2\mathbf{V}^{\mathrm{H}}(\mathbf{V}\mathbf{W} - \mathbf{I}) + 2\lambda\mathbf{W} = 0 \quad \Leftrightarrow \quad (\mathbf{V}^{\mathrm{H}}\mathbf{V} + \lambda\mathbf{I})\mathbf{W} = \mathbf{V}^{\mathrm{H}} \tag{35}$$

$$\Leftrightarrow \quad \mathbf{W} = (\mathbf{V}^{\mathrm{H}}\mathbf{V} + \lambda\mathbf{I})^{-1}\mathbf{V}^{\mathrm{H}} = \mathbf{V}^{\mathrm{H}}(\mathbf{V}\mathbf{V}^{\mathrm{H}} + \lambda\mathbf{I})^{-1}.$$

Последнее равенство доказывается умножением на $(\mathbf{V}\mathbf{V}^{\mathrm{H}} + \lambda\mathbf{I})$ справа и на $(\mathbf{V}^{\mathrm{H}}\mathbf{V} + \lambda\mathbf{I})$ слева. □

Замечание 4. *Алгоритм* $\mathbf{W}_{RZF}(\mathbf{V})$ *также является решением задачи оптимизации с ограничением* (27) *при допущении* $\mathbf{GH} = \mathbf{V}$ *(см. [4]):*

$$\mathbf{W}_{RZF}(\mathbf{V}) = argmin_{\mathbf{W}} \mathbb{E}_{\mathbf{x},\mathbf{n}}\left[\|\mathbf{VWx} - \mathbf{x} + \mathbf{n}\|^2\right], \quad : \|\mathbf{W}\|^2 \leqslant P. \tag{36}$$

Эта задача сводится к (34) *с множителем Лагранжа* $\lambda = L\sigma^2/P$. *Иными словами,* $\mathbf{W}_{RZF}(\mathbf{V})$ *— это частный случай префильтра Винера [4, Eq. (34),(35)], при допущении, что ковариационные матрицы сигнала и шума равны:* $\mathbf{r_x} := \mathbb{E}[\mathbf{xx}^{\mathrm{H}}] = \mathbf{I}_L$ *и* $\mathbf{r_n} := \mathbb{E}[\mathbf{nn}^{\mathrm{H}}] = \sigma^2\mathbf{I}_R$, *а также при специальном выборе постфильтра:* $\mathbf{G} = \mathbf{U}$.

2.2.4. Префильтр Винера с занулением помех

В работе [4] авторы рассматривают регуляризованный псевдообратный префильтр Винера (РПФВ, Wiener Regularized Zero-Forcing, WRZF), который обеспечивает оптимум в задаче (36) для произвольных ковариационных матриц символов и шума $\mathbf{r_x}$ и $\mathbf{r_n}$ при известной матрице постфильтра $\mathbf{G}$. Мы рассматриваем детектирование сопряжённым префильтром (4), т.е. $\mathbf{G} = \mathbf{G}^C$, и применяем префильтр Винера к нормализованному каналу $\mathbf{V}$ при $\mathbf{r_x} = \mathbf{I}_L$ и соответствующей ковариации шума $\mathbf{r_n} = \mathbf{S}^{-2}$.

Алгоритм

$$\mathbf{W}_{WRZF}(\mathbf{V},\mathbf{S}) = \mu\mathbf{V}^{\mathrm{H}}(\mathbf{VV}^{\mathrm{H}} + \lambda\mathbf{I})^{-1}, \quad \lambda = \frac{\sigma^2}{P} tr(\mathbf{S}^{-2}) \tag{37}$$

является решением задачи оптимизации с ограничением (27) при допущении сопряжённого постфильтра ($\mathbf{G} = \mathbf{G}^C$, тогда $\mathbf{G}^C\mathbf{H} = \mathbf{V}$ согласно Теореме 1):

$$\mathbf{W}_{WRZF}(\mathbf{V},\mathbf{S}) = argmin_{\mathbf{W}} \mathbb{E}_{\mathbf{x},\mathbf{n}}\left[\|\mathbf{VWx} - \mathbf{x} + \mathbf{n}\|^2\right], \quad : \|\mathbf{W}\|^2 \leqslant P, \tag{38}$$

где $\mathbb{E}[\mathbf{nn}^{\mathrm{H}}] = \mathbf{S}^{-2}$, $\mathbb{E}[\mathbf{xx}^{\mathrm{H}}] = \mathbf{I}_L$.

2.3. Предлагаемые методы префильтра

2.3.1. Определение адаптивного РПФ

Алгоритмы $\mathbf{W}_{RZF}$ и $\mathbf{W}_{WRZF}$ (а также префильтр Винера в более общем случае) являются алгоритмами построения префильтра со *скалярной регуляризацией*. Учитывая эффективный шум из следствия 1, рассматривается алгоритм *адаптивного РПФ (АРПФ)* с *диагональной матрицей регуляризации*:

$$\mathbf{W}_{ARZF}(\mathbf{V}) = \mu\mathbf{V}^{\mathrm{H}}(\mathbf{VV}^{\mathrm{H}} + \lambda\mathbf{S}^{-2})^{-1}, \quad \lambda = \frac{\sigma^2 L}{P}. \tag{39}$$

АРПФ позволяет использовать различную регуляризацию для разных пользователей и слоёв в зависимости от их сингулярных значений, включая потери пути до пользователя.

Используя $\mathbf{r} = \widetilde{\mathbf{V}}\mathbf{W}\mathbf{x} + \widetilde{\mathbf{S}}^{-1}\widetilde{\mathbf{U}}\mathbf{n}$, запишем квадрат ошибки между принятыми и переданными символами $\mathbf{r}$ и $\mathbf{x}$:

$$\begin{aligned}\|\mathbf{r}-\mathbf{x}\|^2 &= \|\widetilde{\mathbf{V}}\mathbf{W}\mathbf{x}-\mathbf{x}+\widetilde{\mathbf{S}}^{-1}\widetilde{\mathbf{U}}\mathbf{n}\|^2 \\ &\Rightarrow \mathbb{E}_{\mathbf{n}\sim\mathcal{CN}(0,\sigma^2\mathbf{I})}\|(\widetilde{\mathbf{V}}\mathbf{W}-\mathbf{I})\mathbf{x}+\widetilde{\mathbf{S}}^{-1}\widetilde{\mathbf{U}}\mathbf{n}\|^2 = \|(\widetilde{\mathbf{V}}\mathbf{W}-\mathbf{I})\mathbf{x}\|^2 \\ &\Rightarrow \mathbb{E}_{\mathbf{x}\sim\mathcal{CN}(0,\mathbf{I})}\|(\widetilde{\mathbf{V}}\mathbf{W}-\mathbf{I})\mathbf{x}\|^2 = \|\widetilde{\mathbf{V}}\mathbf{W}-\mathbf{I}\|^2. \quad (40)\end{aligned}$$

Введём в определение нормы обратную ковариационную матрицу шума $\widetilde{\mathbf{S}}$, и получим следующую функцию взвешенного метода наименьших квадратов (41):

$$\mathrm{MSE}_S(\mathbf{W}) = \|\widetilde{\mathbf{V}}\mathbf{W}-\mathbf{I}\|^2_{\widetilde{\mathbf{S}}} + \lambda\|\mathbf{W}\|_2^2 = \|\widetilde{\mathbf{S}}(\widetilde{\mathbf{V}}\mathbf{W}-\mathbf{I})\|_2^2 + \lambda\|\mathbf{W}\|_2^2. \quad (41)$$

Таким образом, АРПФ является решением задачи оптимизации (41) (см. теорему 4).

Теорема 4. *Рассмотрим разложение канала $\mathbf{H} = \mathbf{U}^{\mathrm{H}}\mathbf{S}\mathbf{V}$ из леммы 1. Префильтр (39) является решением следующей задачи оптимизации (взвешенная среднеквадратичная ошибка с регуляризацией):*

$$\mathbf{W}_{ARZF}(\mathbf{V}) = argmin_{\mathbf{W}} J_{\mathbf{S}}(\mathbf{W}), \qquad J_{\mathbf{S}}(\mathbf{W}) := \|\mathbf{S}(\mathbf{V}\mathbf{W}-\mathbf{I})\|^2 + \lambda\|\mathbf{W}\|^2. \quad (42)$$

Доказательство. Вычислим градиент и приравняем его к нулю:

$$\begin{aligned}\nabla J_{\mathbf{S}}(\mathbf{W}) = 2\mathbf{V}^{\mathrm{H}}\mathbf{S}(\mathbf{S}\mathbf{V}\mathbf{W}-\mathbf{S})+2\lambda\mathbf{W} = 0 \quad &\Leftrightarrow \quad (\mathbf{V}^{\mathrm{H}}\mathbf{S}^2\mathbf{V}+\lambda\mathbf{I})\mathbf{W} = \mathbf{V}^{\mathrm{H}}\mathbf{S}^2, \\ \mathbf{W} = (\mathbf{V}^{\mathrm{H}}\mathbf{S}^2\mathbf{V}+\lambda\mathbf{I})^{-1}\mathbf{V}^{\mathrm{H}}\mathbf{S}^2 &= \mathbf{V}^{\mathrm{H}}(\mathbf{V}\mathbf{V}^{\mathrm{H}}+\lambda\mathbf{S}^{-2})^{-1}. \quad (43)\end{aligned}$$

Последнее равенство можно доказать, умножив с правой стороны на матрицу $(\mathbf{V}\mathbf{V}^{\mathrm{H}}+\lambda\mathbf{S}^{-2})$ и с левой — на $(\mathbf{V}^{\mathrm{H}}\mathbf{S}^2\mathbf{V}+\lambda\mathbf{I})$. □

Замечание 5. *Задача оптимизации (42) не является стандартной и не может быть записана в форме, аналогичной (36). Тем не менее, такая постановка задачи является более точным приближением к задаче максимизации суммарной спектральной эффективности (25), так как АРПФ обеспечивает более высокую суммарную спектральную эффективность по сравнению с РПФ и префильтром Винера. Подробные результаты моделирования приведены ниже.*

Замечание 6. *Рассматриваемый алгоритм* $\mathbf{W}_{ARZF}(\mathbf{V})$ *и теорема 4 являются основным результатом данной работы. Алгоритмы типа префильтра Винера [4] используют скалярную регуляризацию вида* $\lambda\mathbf{I}$*, поэтому АРПФ не относится к этому классу. Как показано в [5], максимум функции отношения сигнал/шум+помехи (ОСИШ) пользователя (включая суммарную спектральную эффективность (19)) достигается алгоритмом с подходящей диагональной регуляризацией, а АРПФ является эвристикой такого вида.*

Возможная интерпретация функции $J_{\mathbf{S}}(\mathbf{W})$ (34) заключается в следующем. Второй член $\lambda\|\mathbf{W}\|^2$ представляет собой стандартную регуляризацию по шуму, а первый член — норма $\|\mathbf{S}(\mathbf{VW}-\mathbf{I})\|^2$, взвешенная по матрице $\mathbf{S}$, — придаёт больший вес слоям с большими сингулярными значениями. Таким образом, функция оптимизируется более точно для слоёв с более высоким качеством сигнала по сравнению с теми, у которых сигнал слабее. Иными словами, вектора префильтра для слоёв с высокими сингулярными значениями приближаются к псевдообратному префильтру, а для слоёв с низкими — к передаче с сопряжённым префильтром, то есть *адаптивный регуляризованный префильтр обеспечивает адаптивную регуляризацию.* В следующем разделе будет показано, что такой подход приводит к равномерному увеличению спектральной эффективности по сравнению с базовым методом с единичными весами.

Рассмотрим связь адаптивного регуляризационного префильтра с другими алгоритмами. Во-первых, видно, что параметр регуляризации в регуляризационном префильтре Винера представляет собой (арифметическое) среднее значение регуляризации АРПФ:

$$\frac{\sigma^2}{P}tr(\mathbf{S}^{-2}) = \frac{\sigma^2 L}{P}\cdot\frac{1}{L}\sum_{l=1}^{L}s_l^{-2}.$$

В случае, когда потери мощности в канале всех пользователей примерно одинаковы $s_l \approx s, l = 1, ..., L$, АРПФ и префильтр Винера дают схожие результаты. Во-вторых, связь между префильтрами $\mathbf{W}_{RZF}$ и $\mathbf{W}_{ARZF}$ формулируется следующим образом:

Теорема 5. *Пусть матрица канала имеет вид* $\mathbf{H} = \mathbf{U}^{\mathrm{H}}\mathbf{SV}$ *(см. Лемму 1), и обозначим* $\mathbf{F} = \mathbf{UH} = \mathbf{SV}$*. Тогда для АРПФ выполняется следующее соотношение:*

$$\mathbf{W}_{ARZF}(\mathbf{V}) = \mathbf{W}_{RZF}(\mathbf{F})\mathbf{S}. \tag{44}$$

Доказательство.

$$\mathbf{W}_{RZF}(\mathbf{F})\mathbf{S} = \mathbf{F}^{\mathrm{H}}(\mathbf{F}\mathbf{F}^{\mathrm{H}} + \lambda\mathbf{I})^{-1}\mathbf{S} = \mathbf{V}^{\mathrm{H}}\mathbf{S}(\underbrace{\mathbf{S}\mathbf{V}\mathbf{V}^{\mathrm{H}}\mathbf{S} + \lambda\mathbf{I}}_{\mathbf{B}})^{-1}\mathbf{S} =$$
$$= \mathbf{V}^{\mathrm{H}}\mathbf{S}\mathbf{S}^{-1}(\mathbf{V}\mathbf{V}^{\mathrm{H}} + \mathbf{S}^{-1}\lambda\mathbf{I}\mathbf{S}^{-1})^{-1}\mathbf{S}^{-1}\mathbf{S} =$$
$$= \mathbf{V}^{\mathrm{H}}(\underbrace{\mathbf{V}\mathbf{V}^{\mathrm{H}} + \lambda\mathbf{S}^{-2}}_{\mathbf{A}})^{-1} = \mathbf{W}_{ARZF}(\mathbf{V}). \quad (45)$$

□

Замечание 7. *В уравнении (45) акцент сделан на матрицах* $\mathbf{A} = \mathbf{V}\mathbf{V}^{\mathrm{H}} + \lambda\mathbf{S}^{-2}$ *и* $\mathbf{B} = \mathbf{S}\mathbf{V}\mathbf{V}^{\mathrm{H}}\mathbf{S} + \lambda\mathbf{I}$*, чтобы использовать их в следующем утверждении 6.*

Замечание 8. *Формулу АРПФ (39) можно также получить с помощью PCA-разложения [31], что и утверждается в Теореме 4.*

Правая матрица $\mathbf{S}$ в формуле (44) может интерпретироваться как особый вид алгоритма *распределения мощности* (см. интересное исследование в [12, разд. 7]), который распределяет общую мощность передачи между слоями. На практике предпочтительнее использовать $\mathbf{W}_{RZF}(\mathbf{V})$, а не $\mathbf{W}_{RZF}(\mathbf{F})$, так как нормы строк матрицы $\mathbf{F}\mathbf{F}^{\mathrm{H}} + \lambda\mathbf{I}$ могут существенно отличаться (вплоть до нескольких порядков), что приводит к несбалансированному распределению мощности между символами (в качестве альтернативы можно применить корректное распределение мощности для $\mathbf{W}_{RZF}(\mathbf{F})$, как это вытекает из теоремы 4). С другой стороны, параметр регуляризации в $\mathbf{W}_{RZF}(\mathbf{F})$ является более естественным и точным. Исследуемый префильтр $\mathbf{W}_{ARZF}(\mathbf{V})$ объединяет преимущества этих двух подходов и реализует их обобщение.

В данной работе мы доказываем теоретические оценки числа обусловленности обратных ковариационных матриц методов АРПФ и стандартного РПФ, что важно для систем с фиксированной точностью вычислений.

Теорема 6. *Пусть* $\mathbf{V}^{\mathrm{H}}\mathbf{V}(\varepsilon) = \mathbf{I} + O(\varepsilon)$, $\varepsilon \to 0$, *и заданы матрицы* $\mathbf{A} = \mathbf{V}^{\mathrm{H}}\mathbf{V}(\varepsilon) + \lambda\mathbf{S}^{-2} \to \mathbf{I} + \lambda\mathbf{S}^{-2}$ *и* $\mathbf{B} = \mathbf{S}\mathbf{V}^{\mathrm{H}}\mathbf{V}(\varepsilon)\mathbf{S} + \lambda\mathbf{I} \to \mathbf{S}^2 + \lambda\mathbf{I}$*, которые инвертируются в соответствующих префильтрах* $\mathbf{W}_{ARZF} = \mathbf{V}^{\mathrm{H}}\mathbf{A}^{-1}$, *и* $\mathbf{W}_{RZF} = \mathbf{S}\mathbf{V}^{\mathrm{H}}\mathbf{B}^{-1}$. *Тогда числа обусловленности матриц* $\mathbf{A}$ *и* $\mathbf{B}$ *равны соответственно:*

$$\chi(\mathbf{A}) = \frac{\lambda s_{\min}^{-2} + 1}{\lambda s_{\max}^{-2} + 1}, \quad \chi(\mathbf{B}) = \frac{\lambda + s_{\max}^2}{\lambda + s_{\min}^2} \quad (46)$$

и между ними выполняется соотношение:

1) $\chi(\mathbf{A}) < \chi(\mathbf{B})$, *если* $\lambda < s_{\min}^2 < s_{\max}^2$,

2) $\chi(\mathbf{A}) > \chi(\mathbf{B})$, *если* $s_{\min}^2 < s_{\max}^2 < \lambda$,

где $s_{\min}$ *и* $s_{\max}$ *— минимальные и максимальные элементы диагональной матрицы* $\mathbf{S}$, *а* $\lambda = \frac{\sigma^2 L}{P}$.

Доказательство. Предположение о том, что матрица сингулярных векторов пользователей близка к унитарной, справедливо при отборе пользователей с низкой корреляцией: $\mathbf{V}^{\mathrm{H}}\mathbf{V}(\varepsilon) = \mathbf{I} + O(\varepsilon)$, где $\varepsilon \to 0$. Тогда исследуемые матрицы приближаются к диагональным:

$$\mathbf{A} = \mathbf{V}^{\mathrm{H}}\mathbf{V}(\varepsilon) + \lambda\mathbf{S}^{-2} = \mathbf{I} + O(\varepsilon) + \lambda\mathbf{S}^{-2} \to \mathbf{I} + \lambda\mathbf{S}^{-2}.$$

$$\mathbf{B} = \mathbf{S}\mathbf{V}^{\mathrm{H}}\mathbf{V}(\varepsilon)\mathbf{S} + \lambda\mathbf{I} = \mathbf{S}(\mathbf{I} + O(\varepsilon))\mathbf{S} + \lambda\mathbf{I} = \mathbf{S}^2 + \lambda\mathbf{I} + O(\varepsilon)\mathbf{S}^2 \to \mathbf{S}^2 + \lambda\mathbf{I}.$$

Их функции обусловленности соответствуют выражению (46).

Сравнивая эти функции, можно выделить переходные режимы, в которых предпочтительнее использовать ту или иную формулу. Если $\lambda < s_{\min}^2 < s_{\max}^2$, то $\chi(\mathbf{A}) < \chi(\mathbf{B})$, и лучше использовать первую формулу. Если $s_{\min}^2 < s_{\max}^2 < \lambda$, то $\chi(\mathbf{A}) > \chi(\mathbf{B})$, и предпочтительна вторая. □

Замечание 9. *В случае* $s_{\min}^2 < \lambda < s_{\max}^2$ *нельзя сделать однозначный вывод — требуется дополнительный анализ. Однако отметим, что в реальных сетях используется только случай 1)* $\lambda < s_{\min}^2 < s_{\max}^2$. *Сингулярные векторы, чьи сингулярные значения меньше мощности шума, в системе не применяются, поэтому условие* $\chi(\mathbf{A}) < \chi(\mathbf{B})$ *всегда выполняется.*

Математическая формулировка $\mathbf{W}_{ARZF} = \mathbf{V}^{\mathrm{H}}\mathbf{A}^{-1}$ улучшает обусловленность системы при низком и среднем уровне шума λ, что делает алгоритм более точным при реализации в условиях ограниченной разрядной сетки по сравнению с другой формулировкой метода $\mathbf{W}_{RZF} = \mathbf{S}\mathbf{V}^{\mathrm{H}}\mathbf{B}^{-1}$. Экспериментальное сравнение значения обусловленности показано на рис. 4.

2.3.2. Асимптотические свойства алгоритма *АРПФ*

В данном разделе получена асимптотическая оценка функции ОСИШ (16) для префильтра АРПФ (39) при допущении, что мощность шума на каждом устройстве значительно меньше мощности принимаемого сигнала.

Используя разложение по формуле Неймана [32], формулируем следующую лемму

Лемма 3 (Разложение обратной суммы матриц по малому параметру с остаточным членом в форме $\mathcal{O}$)**.** *Рассмотрим обратимые комплексные матрицы* $\mathbf{M}$ *и* $\mathbf{n}$ *одинакового размера и ранга. Для любого* $0 < \varepsilon \ll 1$ *и* $\det\mathbf{M} \neq 0$ *справедливо следующее тождество:*

$$(\mathbf{M} + \varepsilon\mathbf{n})^{-1} = \mathbf{M}^{-1} - \varepsilon\mathbf{M}^{-1}\mathbf{n}\mathbf{M}^{-1} + \mathcal{O}(\varepsilon^2) = \mathbf{M}^{-1} + \mathcal{O}(\varepsilon).$$

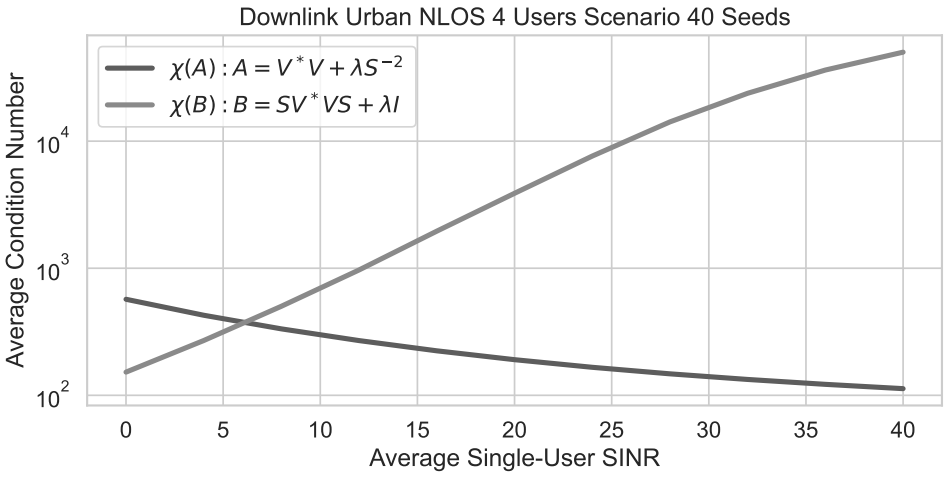

Рис. 4. Числа обусловленности $\chi(\mathbf{A})$ и $\chi(\mathbf{B})$ матриц $\mathbf{A}$ и $\mathbf{B}$.

Доказательство. Рассмотрим разложение матричной функции $\mathbf{F}(\varepsilon)$, зависящей от вещественного параметра $\varepsilon > 0$:

$$\mathbf{F}(\varepsilon) = \mathbf{F}(0) + \mathbf{F}'(0)\varepsilon + \mathcal{O}(\varepsilon^2), \quad \text{где } \mathbf{F}(\varepsilon) = (\mathbf{M} + \varepsilon\mathbf{n})^{-1}.$$

Производная от обратной матрицы приведена в [33, 2.2]:

$$\mathbf{F}'(\varepsilon) = -(\mathbf{M} + \varepsilon\mathbf{n})^{-1}\mathbf{n}(\mathbf{M} + \varepsilon\mathbf{n})^{-1},$$

$$\Rightarrow \mathbf{F}'(0) = -\mathbf{M}^{-1}\mathbf{n}\mathbf{M}^{-1}$$

$$\Rightarrow \mathbf{F}(\varepsilon) = (\mathbf{M} + \varepsilon\mathbf{n})^{-1} = \mathbf{M}^{-1} - \varepsilon\mathbf{M}^{-1}\mathbf{n}\mathbf{M}^{-1} + \mathcal{O}(\varepsilon^2) = \mathbf{M}^{-1} + \mathcal{O}(\varepsilon).$$

□

Следующая теорема 7 описывает асимптотику АРПФ:

Теорема 7. *Пусть матрица* $\mathbf{V}\mathbf{V}^{\mathrm{H}}$ *имеет полный ранг, равный* L*. Тогда выполняются следующие асимптотики:*

$$\mathbf{W}_{ARZF} = \mathbf{V}^{\mathrm{H}}\left((\mathbf{V}\mathbf{V}^{\mathrm{H}})^{-1} - \lambda(\mathbf{V}\mathbf{V}^{\mathrm{H}})^{-1}\mathbf{S}^{-2}(\mathbf{V}\mathbf{V}^{\mathrm{H}})^{-1} + \mathcal{O}(\lambda^2)\right) = \\ = \mathbf{W}_{ZF} - \lambda\mathbf{W}_{ZF}\mathbf{S}^{-2}(\mathbf{V}\mathbf{V}^{\mathrm{H}})^{-1} + \mathcal{O}(\lambda^2), \quad \textit{при } \lambda \to 0+, \quad (47)$$

$$\mathbf{W}_{ARZF} = \lambda^{-1}\mathbf{V}^{\mathrm{H}}\left(\mathbf{S}^2 - \lambda^{-1}\mathbf{S}^2(\mathbf{V}\mathbf{V}^{\mathrm{H}})\mathbf{S}^2 + \mathcal{O}(\lambda^{-2})\right) = \\ = \lambda^{-1}\mathbf{W}_{MRT}\mathbf{S}^2 - \lambda^{-2}\mathbf{W}_{MRT}\mathbf{S}^2(\mathbf{V}\mathbf{V}^{\mathrm{H}})\mathbf{S}^2 + \mathcal{O}(\lambda^{-3}), \quad \textit{при } \lambda \to +\infty. \quad (48)$$

Доказательство. Для получения асимптотики при $\lambda \to 0+$ (47) используем Лемму 3 с $\mathbf{M} = \mathbf{V}\mathbf{V}^{\mathrm{H}}$ и $\mathbf{n} = \mathbf{S}^{-2}$, а асимптотика при $\lambda \to +\infty$ (48) следует из применения Леммы 3 с $\mathbf{M} = \mathbf{S}^{-2}$, $\mathbf{n} = \mathbf{V}\mathbf{V}^{\mathrm{H}}$, $\varepsilon = \lambda^{-1}$. □

Свойство при малом шуме в Теореме 7 при $\lambda \to 0+$ означает, что алгоритм АРПФ стремится к ПФ. Другая асимптотика при $\lambda \to +\infty$ означает, что при преобладании шума над сигналом АРПФ обслуживает только UE с наилучшим каналом благодаря множителю $\mathbf{S}^2$.

Лемма 4 (Модель системы с учётом префильтра АРПФ)**.** *Для префильтра* $\mathbf{W} = \mathbf{W}_{АРПФ}\mathbf{P}$*, постфильтра* $\mathbf{G} = \mathbf{P}^{-1}\mathbf{G}^C$ *и матрицы корреляции* $\mathbf{C} = \mathbf{V}\mathbf{V}^{\mathrm{H}} - \mathbf{I} = \mathcal{O}(\lambda)$ *при* $\lambda \to 0$ *для системы* (1) *справедливо:*

$$\mathbf{GHW} = \mathbf{I} - \lambda\mathbf{S}^{-2} + \mathcal{O}(\lambda^2). \tag{49}$$

Доказательство. Используем Формулу 47:

$$\begin{aligned}\mathbf{V}\mathbf{W}_{ARZF} = \mathbf{V}\mathbf{V}^{\mathrm{H}}(\mathbf{V}\mathbf{V}^{\mathrm{H}} + \lambda\mathbf{S}^{-2})^{-1} = \mathbf{I} - \lambda\mathbf{S}^{-2}(\mathbf{V}\mathbf{V}^{\mathrm{H}})^{-1} + \mathcal{O}(\lambda^2) = \\ = \mathbf{I} - \lambda(\mathbf{I} + \mathcal{O}(\lambda))^{-1} + \mathcal{O}(\lambda^2) = \mathbf{I} - \lambda\mathbf{S}^{-2} + \mathcal{O}(\lambda^2).\end{aligned} \tag{50}$$

$$\begin{aligned}\mathbf{GHW} = \mathbf{P}^{-1}\underbrace{\mathbf{G}^C\mathbf{H}}_{=\mathbf{V}}\mathbf{W}_{ARZF}\mathbf{P} = \mathbf{P}^{-1}\mathbf{V}\mathbf{W}_{ARZF}\mathbf{P} = \\ = \mathbf{P}^{-1}\mathbf{I}\mathbf{P} - \lambda\mathbf{P}^{-1}\mathbf{S}^{-2}\mathbf{P} + \mathcal{O}(\lambda^2) = \mathbf{I} - \lambda\mathbf{S}^{-2} + \mathcal{O}(\lambda^2).\end{aligned}$$

□

Замечание 10. *В реальных сетях множество пользователей выбирается с помощью алгоритма планировщика, а число символов каждого пользователя — алгоритмом адаптации ранга [23]. Оба подхода обеспечивают выполнение* $\mathbf{C} = \mathcal{O}(\lambda)$.

Используя Лемму 4, получаем следующую теорему:

Теорема 8 (Функция ОСИШ с учётом префильтра АРПФ)**.** *Для префильтра* $\mathbf{W} = \mathbf{W}_{ARZF}\mathbf{P}$*, постфильтра* $\mathbf{G} = \mathbf{P}^{-1}\mathbf{G}^C$ *и корреляционной матрицы* $\mathbf{C} = \mathbf{V}\mathbf{V}^{\mathrm{H}} - \mathbf{I} = \mathcal{O}(\lambda)$ *функция ОСИШ* (16) *при условии, что мощность шума намного меньше мощности сигнала* $\lambda = \dfrac{\sigma^2}{P} \to 0$ *и при предположении* $P \sim p_l$, $\forall l = 1, \ldots, L$ *имеет следующий асимптотический вид:*

$$\mathrm{SINR}_l(\mathbf{W}, \mathbf{H}_k, \mathbf{G}_l^C, \sigma^2) = \underbrace{\frac{p_l s_l^2}{\sigma^2}}_{\mathcal{O}(\lambda^{-1})} - 2\underbrace{\frac{p_l}{P}}_{\mathcal{O}(1)} + \underbrace{\mathcal{O}\left(\frac{\sigma^2}{P}\right)}_{\mathcal{O}(\lambda)}, \quad \lambda \to 0. \tag{51}$$

Доказательство.

$$\mathrm{SINR}_l(\mathbf{W}, \mathbf{H}_k, \mathbf{G}_l^C, \sigma^2) := \frac{1 - 2\lambda s_l^{-2} + \mathcal{O}(\lambda^2)}{\mathcal{O}(\lambda^4) + \dfrac{\sigma^2}{p_l s_l^2}} =$$

$$= \frac{p_l s_l^2}{\sigma^2}\left(1 - 2\lambda s_l^{-2} + \mathcal{O}(\lambda^2)\right) =$$

$$= \underbrace{\frac{p_l s_l^2}{\sigma^2}}_{\mathcal{O}(\lambda^{-1})} - 2\underbrace{\frac{p_l}{P}}_{\mathcal{O}(1)} + \underbrace{\mathcal{O}\left(\frac{\sigma^2}{P}\right)}_{\mathcal{O}(\lambda)}.$$

□

2.3.3. Оптимальная регуляризация на основе градиентного метода

Алгоритм: Оптимальная регуляризация префильтра R (OPT)
Вход: Канал $\mathbf{H}$ и его разложение $\mathbf{H} = \mathbf{U}^{\mathrm{H}}\mathbf{S}\mathbf{V}$ по Лемме 1, мощность станции P, шум σ^2, число итераций N

1) **Определим** функцию префильтра $\mathbf{W}(\mathbf{R})$ по формуле (54)
2) **Определим** функцию постфильтра $\mathbf{G}(\mathbf{R}) = \mathbf{G}^{MMSE}(\mathbf{W}(\mathbf{R}))$ по формуле (9)
3) **Определим** целевую функцию $J^{SE}(\mathbf{R}) = SE(\mathbf{W}(\mathbf{R}), \mathbf{H}, \mathbf{G}(\mathbf{R}), \sigma^2)$ из (16), (18), (21)
4) **Задаём** точность завершения по градиенту: $\varepsilon_g = 10^{-5}$
5) **Задаём** точность завершения по аргументу и значению функции: $\varepsilon_c = 10^{-9}$
6) **Инициализируем** регуляризацию: $\mathbf{R}_0 = \dfrac{L\sigma^2}{P}\mathbf{S}^{-2}$

7) **Для** $n = 0$ **до** $N - 1$ **выполнять:**

   а) Вычисляем градиент $\nabla_{\mathbf{R}} J^{SE}(\mathbf{R})\big|_{\mathbf{R}=\mathbf{R}_n}$

   б) **Если** выполняются условия сходимости по ε_g или ε_c:

      – **Вернуть** $\mathbf{W}_{OPT} = \mathbf{W}(\mathbf{R}_n)$

   в) **Иначе:**

      – Вычисляем направление L-BFGS [34] по градиенту: $\mathbf{D}_n = \mathbf{D}(\nabla_{\mathbf{R}} J^{SE}(\mathbf{R})\big|_{\mathbf{R}=\mathbf{R}_n})$

      – Оптимизируем длину шага: $\alpha_n = \arg\max\limits_{\alpha} J^{SE}(\mathbf{R}_n + \alpha \mathbf{D}_n)$

      – Обновляем регуляризацию: $\mathbf{R}_{n+1} = \mathbf{R}_n + \alpha_n \mathbf{D}_n$

8) **Вернуть** $\mathbf{W}_{OPT} = \mathbf{W}(\mathbf{R}_N)$

Таблица 2. Оптимальная регуляризация префильтра $\mathbf{R}$ (OPT)

Рассматриваемый алгоритм минимизирует квадратичную оптимизационную задачу (42), однако остаётся вопрос: насколько он хорош относительно функции суммарной спектральной эффективности (21)?

В работе [5] доказано, что оптимальное решение задачи $\max f(SINR_1, \ldots, SINR_K)$ при ограничении по суммарной мощности имеет форму:

$$\mathbf{W}_{OPT}(\mathbf{V}) = \mu \mathbf{V}^{\mathrm{H}} (\mathbf{V}\mathbf{V}^{\mathrm{H}} + \mathbf{r})^{-1}, \quad \mathbf{r} = \mathrm{diag}(r_1, \ldots, r_L). \tag{52}$$

Хотя аналитически определить конкретные значения $\mathbf{r}$ затруднительно, структура решения остаётся полезной.

В связи с этим мы сравниваем эффективность алгоритма АРПФ с градиентным поиском по суммарной функции *спектральной эффективности* (19). Для этого формулируется итеративное решение с дифференцируемыми вложенными функциями. Предлагаемое параметрическое решение сохраняет структуру базового алгоритма РПФ, но оптимизирует целевую функцию *спектральной эффективности*, что приводит к заметному улучшению качества. Мы решаем задачу гладкой оптимизации с ограничениями: требуется найти локальный максимум спектральной эффективности (21):

$$\begin{aligned} &\text{максимизировать}_{\mathbf{R}=\mathrm{diag}\{r_1,\ldots,r_L\}} && \mathrm{SE}(\mathbf{W}(\mathbf{R}), \mathbf{H}, \mathbf{G}^{\mathrm{MMSE}}(\mathbf{W}), \sigma^2) \\ &\text{при условиях} && |(w_{t1}, \ldots, w_{tL})|^2 \leqslant \frac{P}{T}, \quad t = 1, \ldots, T \end{aligned} \tag{53}$$

Параметрическое решение использует формулу РПФ в следующем виде:

$$\mathbf{W}(\mathbf{V},\mathbf{r}) = \mu(\widehat{\mathbf{W}})\widehat{\mathbf{W}}(\mathbf{V},\mathbf{r}), \quad \widehat{\mathbf{W}}(\mathbf{r}) = \mathbf{V}^{\mathrm{H}}(\mathbf{V}\mathbf{V}^{\mathrm{H}} + \mathbf{r})^{-1}, \tag{54}$$

$$\mu(\widehat{\mathbf{W}}) = \frac{\sqrt{P/T}}{\max\limits_{t} \|(\hat{w}_{t1}, \ldots, \hat{w}_{tL})\|} \tag{55}$$

Ограничение на максимальную мощность антенн реализуется через масштабирование матрицы префильтра скаляром μ, что позволяет удовлетворить ограничению мощности, сохранив геометрию и желаемые свойства префильтра.

В дальнейших экспериментах будет оптимизироваться дифференцируемая вещественная диагональная матрица $\mathbf{r} \in \mathbb{R}^{L\times L}$, что составляет одно из основных достижений работы. Сам префильтр $\mathbf{W}$ изучен в соответствующей статье [35].

Процесс постфильтра участвует в расчёте градиента и, следовательно, является неотъемлемой частью спектральной эффективности. Дифференцируемыми переменными выступают диагональные элементы матрицы регуляризации, по которым строится префильтр, затем вычисляется постфильтр. Оба результата используются при расчёте градиента. Регуляризация участвует во всех этих операциях как внутренняя переменная сложной композиционной функции, и её градиент может быть найден по правилу дифференцирования сложной функции с использованием алгоритма обратного распространения ошибки, как это реализовано, например, в библиотеке PyTorch.

Замечание 11. *В рассмотренной градиентной оптимизации префильтр строится с учётом конкретного постфильтра (в данном случае постфильтра среднеквадратичной ошибки), который предполагается реализуемым на стороне пользователя. Эта идея активно обсуждается в современной литературе (см., например, [10]) и может быть применена для улучшения практически любой схемы префильтра с помощью итеративной процедуры.*

3. Результаты моделирования и обсуждение

3.1. Настройка в “Quadriga”

В этом разделе описано, как получены данные с помощью “Quadriga” — открытого ПО для генерации реалистичного радиоканала. Рассматривается сценарий Urban Non-Line-of-Sight (3GPP_38.901_RMa_NLOS) [22].

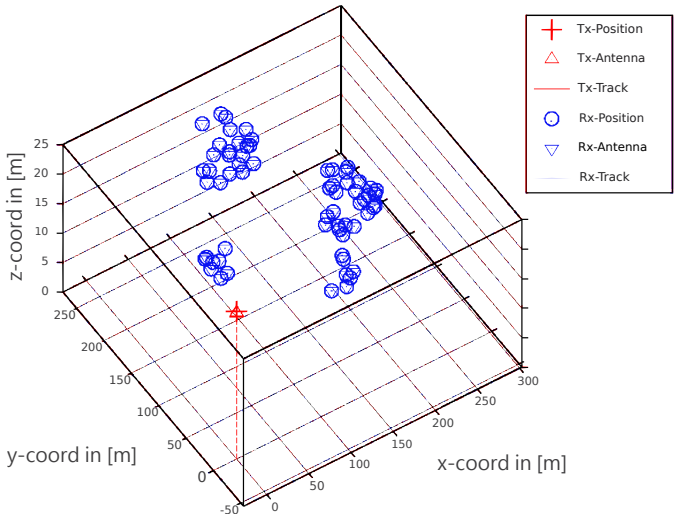

Рис. 5. Пример случайной генерации пользователей в городской среде.

Параметр	**Значение**
Базовая станция	
Число станций	1
Координаты (x,y,z), м	$(0, 0, 25)$
Антенн по оси y/z	8 / 4
Расст. между антеннами (y/z)	0.5λ / 1.7λ
Модель антенны	3gpp-macro
Ширина луча (азимут/место), град	60 / 10
Отн. перед/зад лепестков, дБ	20
Всего антенн	64
Приёмник	
Антенн по оси x	2
Расст. между антеннами (x)	0.5λ
Модель антенны	полуволновый диполь
Всего антенн	4
Симуляция Quadriga	
Центр. частота	3,5 ГГц
Выборка на метр	1
Задержка прямого пути	1
Без сферических волн	1
Генератор каналов	
Затухание в тени (sigma)	0
Деление кластеров	Ложь
Полоса пропускания	100 МГц
Число поднесущих	42

Таблица 3. Параметры генерации канала в “Quadriga”

На рис. 5 показан пример расположения пользователей: они размещаются либо в кластере одного из зданий, либо на земле рядом с ним.

Пользователи обозначены синими кружками, а базовая станция — красным. Расстояния между антеннами станции и пользователями малые по сравнению с расстояниями между станцией и пользователями, поэтому они изображены отдельными кружками, каждый содержащим несколько антенн.

Общий алгоритм моделирования для каждого случайного зерна:

1) Генерируем случайную среду вокруг базовой станции;

2) Случайно размещаем пользователей рядом с базовой станцией;

3) Выбираем релевантных пользователей по корреляции.

Далее процедура описана более подробно.

Во-первых, фиксируем положение базовой станции в точке $[0, 0, 25]$.

Во-вторых, выбираем случайные позиции пользователей вокруг станции:

1) Сэмплируем до 8 центров кластеров (x_c, y_c) в секторе 120° радиусом до 2000 м от станции. Каждый кластер соответствует части здания;

2) Задаём случайную высоту кластера $z_c = 1.5 + (3 \cdot U(\{1, \ldots, 10\}) - 1)$ м, выбирая этаж равномерно;

3) Для каждого пользователя назначаем кластер $c(u)$ и сэмплируем (x_u, y_u) внутри окружности радиуса 60 м вокруг центра кластера;

4) Сэмплируем высоту пользователя: 80% размещаются около этажа кластера $z_u = z_{c(u)} + 3\,U(\{-1, 0, 1\})$, и 20% — на улице $z_u = 1.5$м.

В-третьих, сгенерировав матрицы канала для $K_{\max} = 64$ пользователей, выполняем отбор, моделируя работу планировщика. Выбираем $K < K_{\max}$ пользователей, у которых корреляция не слишком высокая: $corr_{i,j} = |\mathbf{V}_{i,1}^{\mathrm{H}} \mathbf{V}_{j,1}|^2 \leq 0.3$. Число потоков для каждого пользователя $L_k = 2$ (политика выбора ранга).

Рассматриваем два сценария:

1) Разные потери мощности на пути (path loss, PL): введён случайный множитель PL $\in [-10\,\text{дБ}, +10\,\text{дБ}]$ — диапазон, характерный для близких пользователей (в реальности разброс внутри базовой станции до 60 дБ);

2) Почти равные потери мощности на пути: $s_{i,1} \sim s_{j,1}$ (это поведение по умолчанию у “Quadriga”).

Результаты усреднены по 40 моделям каналов и выборок пользователей:

- $\mathbf{H} \in \mathbb{C}^{K \times R_k \times T}$;

- $T = 64$ антенн базовой станции;

- $K = 4$ пользователей;

- $R = 16$ антенн на стороне пользователя;

- $L = 8$ передаваемых символов.

Несущая частота выбирается случайно в пределах полосы. Антенная решётка базовой станции: 8×4 (8 по оси по оси x и 4 по оси y), антенная решётка на пользователе: 2×1. Каждая позиция антенной решётки имеет две кросс-поляризованные антенны. Алгоритм генерации даёт реалистичную городскую среду. Подробные параметры приведены в Табл. 3 приложения.

3.2. Результаты

Сравниваем адаптивный регуляризованный псевдообратный префильтр с опорными алгоритмами (сопряжённый префильтр (MRT), псевдообратный префильтр (ZF), регуляризованный псевдообратный префильтр (RZF), регуляризованный псевдообратный префильтр Винера (WRZF)) и оптимальным решением OPT. На рисс. 6, табл. 4 и рисс. 7, табл. 5 представлены средняя (J_SE) и минимальная спектральная эффективность для сценария с разными потерями мощности на путях. Аналогичные данные для сценария с равными потерями мощности на путях — на рис. 8/табл. 6 и рис. 9/табл. 7.

Префильтр SU SINR	$W_{MRT}(V)$	$W_{ZF}(V)$	$W_{ZF}(F)$	$W_{RZF}(V)$	$W_{RZF}(F)$	$W_{WRZF}(V)$	$W_{ARZF}(V)$	$W_{OPT}(V)$
0	1.45	2.27	0.02	2.37	2.46	1.48	3.91	4.07
4	1.75	3.13	0.05	3.19	2.79	1.84	4.91	5.10
8	2.08	4.16	0.11	4.19	3.10	2.33	6.03	6.32
12	2.41	5.29	0.26	5.31	3.37	3.02	7.17	7.63
16	2.71	6.51	0.54	6.52	3.61	4.02	8.23	9.01
20	2.99	7.79	1.03	7.79	3.66	5.48	9.26	10.42
24	3.24	9.05	1.75	9.05	3.76	7.29	10.23	11.74
28	3.47	10.40	2.77	10.40	4.08	9.58	11.24	13.09
32	3.65	11.65	3.95	11.65	4.73	11.60	12.19	14.17
36	3.76	12.83	5.16	12.83	5.55	13.10	13.13	15.24
40	3.83	13.90	6.34	13.90	6.52	14.21	14.05	16.17

Таблица 4. Таблица соответствует рис. 6 и показывает среднюю спектральную эффективность (SE) различных префильтров в *городском многолучевом (NLOS) сценарии* с *разными потерями пути*. Исследуемый алгоритм — $\mathbf{W}_{ARZF}(\mathbf{V})$. Оптимальная регуляризация $\mathbf{W}_{OPT}(\mathbf{V})$ была получена с помощью оптимизации методом L-BFGS.

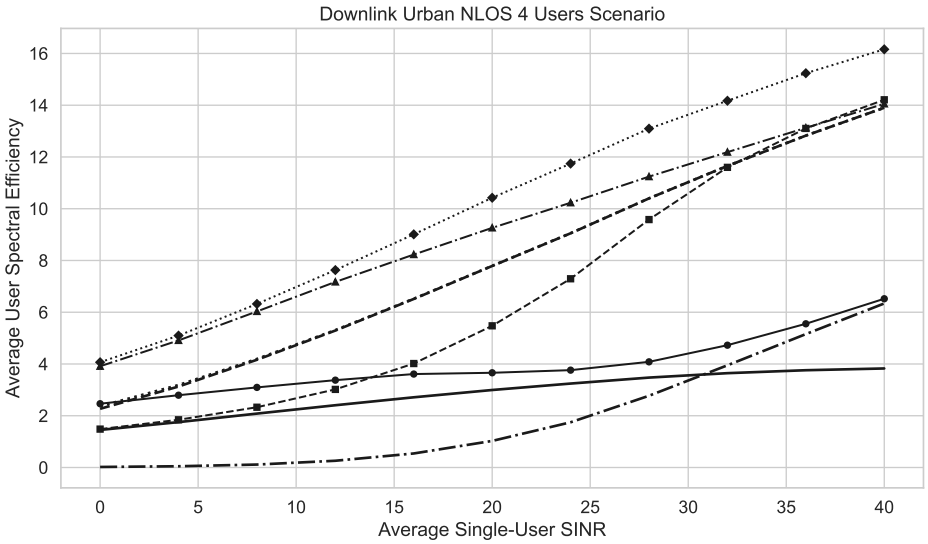

Рис. 6. Средняя спектральная эффективность (SE) различных префильтров в *городском многолучевом (NLOS) сценарии* с *разными потерями пути*. Зелёная линия совпадает с жёлтой. Матрица $\mathbf{F} = \mathbf{SV}$. Значения приведены в табл. 4.

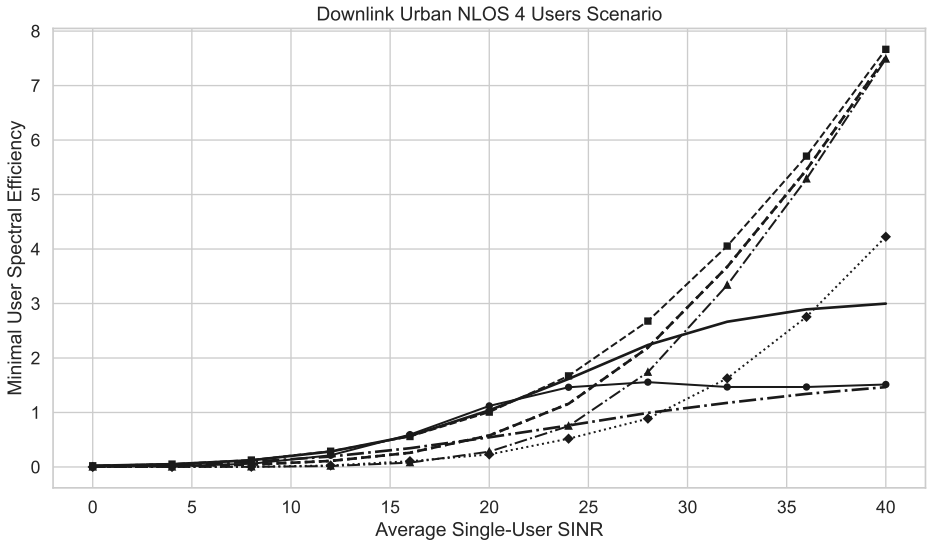

Рис. 7. Минимальная спектральная эффективность (SE) среди пользователей для различных префильтров в *городском многолучевом (NLOS) сценарии* с *разными потерями пути*. Зелёная линия совпадает с жёлтой. Матрица $\mathbf{F} = \mathbf{SV}$. Значения приведены в табл. 5.

Префильтр SU SINR	$W_{MRT}(V)$	$W_{ZF}(V)$	$W_{ZF}(F)$	$W_{RZF}(V)$	$W_{RZF}(F)$	$W_{WRZF}(V)$	$W_{ARZF}(V)$	$W_{OPT}(V)$
0	0.02	0.01	0.02	0.01	0.00	0.02	0.00	0.00
4	0.05	0.02	0.04	0.02	0.01	0.05	0.00	0.00
8	0.12	0.04	0.10	0.04	0.06	0.13	0.00	0.00
12	0.28	0.11	0.19	0.11	0.21	0.29	0.02	0.03
16	0.57	0.26	0.34	0.26	0.59	0.56	0.08	0.11
20	1.04	0.58	0.54	0.58	1.12	1.01	0.28	0.23
24	1.62	1.16	0.76	1.16	1.46	1.67	0.75	0.52
28	2.24	2.19	0.99	2.19	1.56	2.68	1.74	0.89
32	2.67	3.67	1.18	3.67	1.47	4.05	3.34	1.63
36	2.89	5.45	1.34	5.45	1.47	5.70	5.29	2.74
40	3.00	7.52	1.47	7.52	1.51	7.67	7.49	4.25

Таблица 5. Таблица соответствует рис. 7 и показывает минимальную спектральную эффективность (SE) различных префильтров в *городском многолучевом (NLOS) сценарии* с *разными потерями пути*. Исследуемый алгоритм — $\mathbf{W}_{ARZF}(\mathbf{V})$. Оптимальная регуляризация $\mathbf{W}_{OPT}(\mathbf{V})$ была получена с помощью оптимизации методом L-BFGS.

Адаптивный регуляризованный псевдообратный префильтр показывает лучшую среднюю спектральную эффективность: адаптивная регуляризация учитывает потери мощности на пути и порядок сингулярных значений каждого пользователя. В диапазоне высоких одномользовательских ОСИШ псевдообратный префильтр лучше сопряжённого префильтра. При равных потерях мощности на путях спектральная эффективность АРПФ примерно равена спектральной эффективности префильтру Винера; при разных потерях мощности на путях префильтр Винера заметно хуже, а АРПФ выигрывает.

OPT (чёрная линия) — градиентный поиск — даёт лучшие результаты, но слишком затратен вычислительно. Тем не менее он полезен как верхняя оценка: он показывает потенциал улучшения АРПФ. Анализ минимальной спектральной эффективности показывает, что АРПФ улучшает среднюю спектральную эффективность, но слабее для самых слабых пользователей, особенно при низком однопользовательском ОСИШ. Такая разница между средней и минимальной сапектральной эффективностью известна [12].

На рис. 8/табл. 6 и рис. 9/табл. 7 представлены аналогичные результаты для равных потерь мощности на пути : АРПФ превосходит по средней спекральной эффективности, но не по минимальной; при этом он совпадает с префильтром Винера, когда потери мощности на пути одинаковые.

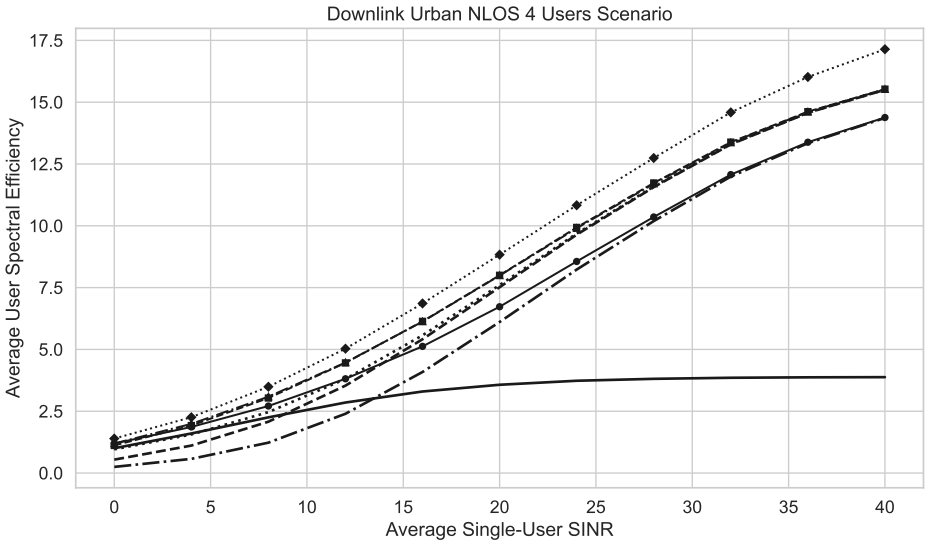

Рис. 8. Средняя спектральная эффективность (SE) различных префильтров в *городском многолучевом (NLOS) сценарии* с *равными потерями пути* (см. табл. 6). Красная линия $\mathbf{W}_{ARZF}(\mathbf{V})$ совпадает с синей линией $\mathbf{W}_{WRZF}(\mathbf{V})$, что говорит о равенстве работы алгоритмов при одинаковых потерях.

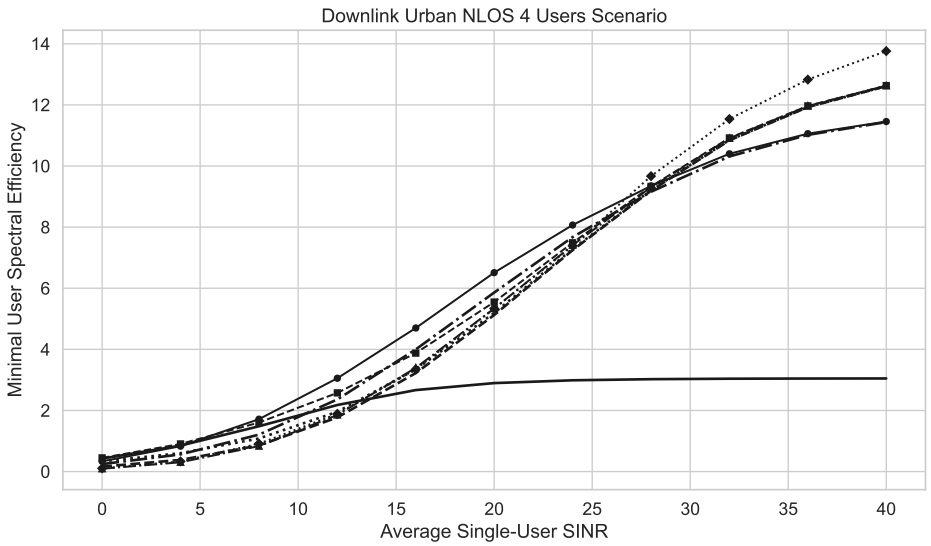

Рис. 9. Минимальная спектральная эффективность (SE) среди пользователей для различных префильтров в *городском многолучевом (NLOS) сценарии* с *равными потерями пути* (см. табл. 7).

Префильтр *SUSINR*	$W_{MRT}(V)$	$W_{ZF}(V)$	$W_{ZF}(F)$	$W_{RZF}(V)$	$W_{RZF}(F)$	$W_{WRZF}(V)$	$W_{ARZF}(V)$	$W_{OPT}(V)$
0	1.00	0.54	0.25	0.96	1.19	1.12	1.19	1.39
4	1.60	1.11	0.57	1.56	1.86	1.92	1.99	2.26
8	2.25	2.08	1.23	2.46	2.71	3.03	3.09	3.49
12	2.85	3.54	2.40	3.81	3.81	4.46	4.47	5.03
16	3.30	5.41	4.09	5.57	5.12	6.14	6.12	6.86
20	3.57	7.52	6.11	7.60	6.73	8.00	7.99	8.83
24	3.73	9.64	8.23	9.69	8.56	9.94	9.90	10.83
28	3.81	11.56	10.19	11.59	10.36	11.73	11.70	12.74
32	3.85	13.29	11.99	13.31	12.07	13.38	13.36	14.59
36	3.87	14.57	13.34	14.58	13.38	14.62	14.60	16.02
40	3.88	15.50	14.36	15.51	14.38	15.53	15.52	17.15

Таблица 6. Таблица соответствует рис. 8 и показывает среднюю спектральную эффективность различных префильтров в *городском многолучевом (NLOS) сценарии*. Исследуемый алгоритм — $\mathbf{W}_{ARZF}(\mathbf{V})$. Оптимальная регуляризация $\mathbf{W}_{OPT}(\mathbf{V})$ была настроена с помощью алгоритма оптимизации L-BFGS.

Префильтр *SUSINR*	$W_{MRT}(V)$	$W_{ZF}(V)$	$W_{ZF}(F)$	$W_{RZF}(V)$	$W_{RZF}(F)$	$W_{WRZF}(V)$	$W_{ARZF}(V)$	$W_{OPT}(V)$
0	0.42	0.16	0.24	0.33	0.33	0.45	0.09	0.11
4	0.85	0.38	0.56	0.60	0.83	0.91	0.30	0.33
8	1.48	0.86	1.21	1.08	1.71	1.60	0.83	0.91
12	2.18	1.77	2.36	1.96	3.05	2.58	1.85	1.89
16	2.67	3.22	4.00	3.34	4.70	3.88	3.40	3.35
20	2.90	5.12	5.86	5.19	6.51	5.55	5.34	5.33
24	2.99	7.25	7.67	7.28	8.07	7.49	7.41	7.38
28	3.02	9.19	9.16	9.21	9.35	9.33	9.30	9.67
32	3.04	10.84	10.31	10.85	10.40	10.92	10.91	11.54
36	3.04	11.93	11.02	11.93	11.06	11.97	11.96	12.83
40	3.05	12.61	11.44	12.62	11.45	12.63	12.63	13.74

Таблица 7. Таблица соответствует рис. 9 и показывает минимальную спектральную эффективность различных префильтров в *городском многолучевом (NLOS) сценарии*. Исследуемый алгоритм — $\mathbf{W}_{ARZF}(\mathbf{V})$. Оптимальная регуляризация $\mathbf{W}_{OPT}(\mathbf{V})$ была настроена с помощью алгоритма оптимизации L-BFGS.

4. Заключение

Ключевая задача статьи – оптимизация метода построения префильтра в многопользовательской системе с множеством антенн, в том числе на каждом приемном устройстве. В статье мы анализируем производительность различных линейных методов (сопряжённый префильтр, псевдообратный префильтр, регуляризованный псевдообратный префильтр, регуляризованный псевдообратный префильтр Винера) и вводим простую эвристическую модель адаптивного регуляризованного псевдообратного префильтра. Аналитически изучаем связь адаптивного регуляризованного псевдообратного префильтра с регуляризованным псевдообратным префильтром, а также их асимптотическое соотношение. Получаем оценку ОСИШ при низком шуме. Проведенное тестирование на реалистичном канале в системе “Quadriga” показывает стабильное улучшение средней (по принимающим устройствам) спектральной эффективности по сравнению с эталонными методами. Важно отметить, что при этом минимальная спектральная эффективность (рассчитываемая на подмножестве принимающих устройств с самым плохим априорным ОСШ) практически не меняется. Результаты для предложенного префильтра также сравниваются с нелинейнымй методом OPT, который градиентным алгоритмом оптимизирует целевую функцию и таким образом дает хорошую оценку для доступной верхней границы решения.

Доступность данных и материалов

Матрицы каналов, сгенерированные в “Quadriga”, и полный Python-код доступны на GitHub: *https://github.com/eugenbobrov/Adaptive-Regularized-Zero-Forcing-Beamforming-in-Massive-MIMO-with-Multi-Antenna-Users.*

Adaptive Regularized Zero-Forcing Beamforming in Massive MIMO with Multi-Antenna Users
Bobrov E.A., Minenkov D.S., Yudakov D.A.

Modern cellular networks utilize massive MIMO technology with multiple antennas. This work investigates the adaptive regularized Zero-Forcing method employing special regularization based on SVD. We conduct theoretical analysis, performance evaluation, and comparison with other methods through simulations using the Quadriga channel model.

Keywords: Telecommunications, MIMO, optimization, singular value decomposition (SVD), signal-to-interference-and-noise ratio (SINR), spectral efficiency

References

[1] Ngo, Hien Quoc and Larsson, Erik G and Marzetta, Thomas L, “Energy and spectral efficiency of very large multiuser MIMO systems”, *IEEE Transactions on Communications*, **61**:4 (2013), 1436–1449.

[2] Andrews, Jeffrey G and Buzzi, Stefano and Choi, Wan and Hanly, Stephen V and Lozano, Angel and Soong, Anthony CK and Zhang, Jianzhong Charlie, “What will 5G be?”, *IEEE Journal on selected areas in communications*, **32**:6 (2014), 1065–1082.

[3] Parfait, Tebe, Kuang, Yujun, Jerry, Kponyo, “Performance analysis and comparison of ZF and MRT based downlink massive MIMO systems”, *2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2014, 383–388.

[4] Joham, Michael, Utschick, Wolfgang, Nossek, Josef A., “Linear transmit processing in MIMO communications systems”, *IEEE Transactions on Signal Processing*, **53**:8 (2005), 2700–2712.

[5] Björnson, Emil, Bengtsson, Mats, Ottersten, Björn, “Optimal multiuser transmit beamforming: A difficult problem with a simple solution structure [lecture notes]”, *IEEE Signal Processing Magazine*, **31**:4 (2014), 142–148.

[6] Nguyen, Long D., Tuan, Hoang Duong, Duong, Trung Q., Poor, H. Vincent, "Multi-user regularized zero-forcing beamforming", *IEEE Transactions on Signal Processing*, **67**:11 (2019), 2839–2853.

[7] Zhang, Jiankang, Chen, Sheng, Maunder, Robert G., Zhang, Rong, Hanzo, Lajos, "Regularized zero-forcing precoding-aided adaptive coding and modulation for large-scale antenna array-based air-to-air communications", *IEEE Journal on Selected Areas in Communications*, **36**:9 (2018), 2087–2103.

[8] Peel, Christian B., Hochwald, Bertrand M., Swindlehurst, A. Lee, "A vector-perturbation technique for near-capacity multiantenna multiuser communication-part I: channel inversion and regularization", *IEEE Transactions on Communications*, **53**:1 (2005), 195–202.

[9] Jaeckel, Stephan, Raschkowski, Leszek, Börner, Kai, Thiele, Lars, "QuaDRiGa: A 3-D multi-cell channel model with time evolution for enabling virtual field trials", *IEEE Transactions on Antennas and Propagation*, **62**:6 (2014), 3242–3256.

[10] Shi, Shuying, Schubert, Martin, Boche, Holger, "Downlink MMSE transceiver optimization for multiuser MIMO systems: Duality and sum-MSE minimization", *IEEE Transactions on Signal Processing*, **55**:11 (2007), 5436–5446.

[11] Caire, Giuseppe, Shamai, Shlomo, "On the achievable throughput of a multiantenna Gaussian broadcast channel", *IEEE Transactions on Information Theory*, **49**:7 (2003), 1691–1706.

[12] Björnson, Emil and Hoydis, Jakob and Sanguinetti, Luca, "Massive MIMO networks: Spectral, energy, and hardware efficiency", *Foundations and Trends in Signal Processing*, **11**:3-4 (2017), 154–655.

[13] Tran, Le-Nam and Juntti, Markku and Bengtsson, Mats and Ottersten, Bjorn, "Beamformer designs for MISO broadcast channels with zero-forcing dirty paper coding", *IEEE transactions on wireless communications*, **12**:3 (2013), 1173–1185.

[14] Fatema, Nusrat and Hua, Guang and Xiang, Yong and Peng, Dezhong and Natgunanathan, Iynkaran, "Massive MIMO linear precoding: A survey", *IEEE systems journal*, **12**:4 (2017), 3920–3931.

[15] Dhakal, Sunil, "High rate signal processing schemes for correlated channels in 5G networks", 2019.

[16] Bogale, Tadilo Endeshaw, Vandendorpe, Luc, "Sum MSE optimization for downlink multiuser MIMO systems with per antenna power constraint: Downlink-uplink duality approach", *2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, 2011, 2035–2039.

[17] Wiesel, Ami, Eldar, Yonina C., Shamai, Shlomo, "Zero-forcing precoding and generalized inverses", *IEEE Transactions on Signal Processing*, **56**:9 (2008), 4409–4418.

[18] Hoydis, Jakob and Ten Brink, Stephan and Debbah, Mérouane, "Massive MIMO in the UL/DL of cellular networks: How many antennas do we need?", *IEEE Journal on selected Areas in Communications*, **31**:2 (2013), 160–171.

[19] Björnson, Emil and Jorswieck, Eduard and Ottersten, Bjorn, "Impact of spatial correlation and precoding design in OSTBC MIMO systems", *IEEE Transactions on Wireless Communications*, **9**:11 (2010), 3578–3589.

[20] Sun, Liang, McKay, Matthew R., "Eigen-based transceivers for the MIMO broadcast channel with semi-orthogonal user selection", *IEEE Transactions on Signal Processing*, **58**:10 (2010), 5246–5261.

[21] Nguyen, Duy HN, Le-Ngoc, Tho, "MMSE precoding for multiuser MISO downlink transmission with non-homogeneous user SNR conditions", *EURASIP Journal on Advances in Signal Processing*, **2014**:1 (2014), 1–12.

[22] Tse, David, Viswanath, Pramod, "Fundamentals of wireless communication", 2005.

[23] Mahmood, Nurul H., Berardinelli, Gilberto, Tavares, Fernando ML, Lauridsen, Mads, Mogensen, Preben, Pajukoski, Kari, "An efficient rank adaptation algorithm for cellular MIMO systems with IRC receivers", *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, 2014, 1–5.

[24] Bobrov, Evgeny, Chinyaev, Boris, Kuznetsov, Viktor, Minenkov, Dmitrii, Yudakov, Daniil, "Power allocation algorithms for massive MIMO systems with multi-antenna users", *Wireless Networks*, 2023, 1–22.

[25] Ren, Bin, Wang, Yingmin, Sun, Shaohui, Zhang, Yawen, Dai, Xiaoming, Niu, Kai, "Low-complexity MMSE-IRC algorithm for uplink massive MIMO systems", *Electronics Letters*, **53**:14 (2017), 972–974.

[26] Mehana, Ahmed Hesham, Nosratinia, Aria, "Diversity of MMSE MIMO receivers", *IEEE Transactions on Information Theory*, **58**:11 (2012), 6788–6805.

[27] Lagen, Sandra, Wanuga, Kevin, Elkotby, Hussain, Goyal, Sanjay, Patriciello, Natale, Giupponi, Lorenza, "New radio physical layer abstraction for system-level simulations of 5G networks", *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, 2020, 1–7.

[28] Kuhn, Harold W., Tucker, Albert W., "Nonlinear programming", *Traces and Emergence of Nonlinear Programming*, 2014, 247–258.

[29] Koopmans, Tjalling, "Activity analysis of production and allocation", 1951.

[30] Tran, Le-Nam, "An iterative precoder design for successive zero-forcing precoded systems", *IEEE Communications Letters*, **16**:1 (2011), 16–18.

[31] Pearson, Karl, "On lines and planes of closest fit to systems of points in space", *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, **2**:11 (1901), 559–572.

[32] Zhu, Dengkui, Li, Boyu, Liang, Ping, "On the matrix inversion approximation based on Neumann series in massive MIMO systems", *2015 IEEE International Conference on Communications (ICC)*, 2015, 1763–1769.

[33] Petersen, K. B., Pedersen, M. S., "The Matrix Cookbook", 2008 http://www2.imm.dtu.dk/pubdb/p.php?3274.

[34] Liu, Dong C., Nocedal, Jorge, "On the limited memory BFGS method for large scale optimization", *Mathematical Programming*, **45**:1 (1989), 503–528.

[35] Bobrov, Evgeny, Kropotov, Dmitry, Troshin, Sergey, Zaev, Danila, *arXiv:2107.13440*, 2021.

Бинаризация языковых моделей

Д. Н. Давыдова[1]

В последние годы в сфере обработки естественного языка широкое распространение получили большие языковые модели. Но, несмотря на их востребованность, их применение становится затруднительным из-за больших затрат времени, энергии и памяти.

Одним из способов решения этой проблемы является квантизация нейронных сетей — преобразование весов и активаций сети к представлению с более низкой точностью. Частным случаем квантизации является бинаризация — приведение параметров сети к разрядности 1 бит.

В работе рассмотрена структура бинарных нейронных сетей, приведен обзор текущих методов бинаризации языковых моделей, описаны полученные результаты.

Ключевые слова: обработка естественного языка, бинарные нейронные сети, бинаризация, квантизация, большие языковые модели.

1. Введение

Глубокие нейронные сети позволяют получить высокие результаты в различных задачах, таких, как классификация изображений, распознавание речи, машинный перевод и обработка естественного языка. В последние годы все более востребованными становятся большие языковые модели — языковые модели, содержащие более миллиарда параметров и позволяющие с высокой точностью выполнять различные языковые задачи [2], [1].

Но, несмотря на востребованность и преимущества больших языковых моделей, огромное количество параметров делает их обучение и запуск проблематичным. Обучение больших языковых моделей занимает много времени и требует больших вычислительных мощностей, а запуск таких моделей на устройствах с ограниченным количеством памяти, таких, как мобильные телефоны, оказывается затруднительным. В частности, обучение LLaMA-7B на 1T токенов на GPU A100-80GB заняло 82432 часа и потребовало 36 МВт $\cdot$ ч энергии [2], а ее хранение требует 12.55 ГБ, поэтому работать с такой моделью невозможно без мощных видеокарт.

[1] *Давыдова Дарья Николаевна* — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: d.dawydowa2017@yandex.ru.

Davydova Daria Nikolaevna — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

Для решения этой проблемы предлагались различные методы оптимизации нейронных сетей, то есть, уменьшения затрат по времени или памяти моделей таким образом, чтобы потери в качестве при этом были как можно меньше.

Методы оптимизации можно условно разделить на методы, меняющие архитектуру модели, такие, как удаление или изменение слоев сети [3], [4]; методы сжатия модели, такие, как квантизация [5], [6] и прунинг [7], [8]; метод дистилляции знаний [9].

Квантизация предполагает сжатие модели за счет преобразования параметров и активаций сети к представлению с более низкой точностью, например, к 8, 4, 2 или 1 бит. Такой подход позволяет увеличить пропускную способность нейронной сети и упростить хранение модели, не изменяя при этом ее архитектуру.

Частным случаем квантизации является бинаризация нейронных сетей. При бинаризации нейронных сетей веса и активации, изначально представленные с точностью 32 бит, заменяются 1-битным представлением, а операции умножения матриц, затрачивающие большое количество вычислительных ресурсов, заменяются на булевы операции. Таким образом можно существенно ускорить выполнение вычислений модели и упростить ее хранение.

В данном обзоре будут рассмотрены результаты исследований по применению метода бинаризации к большим языковым моделям и проведен анализ полученных результатов.

2. Структура бинарных нейронных сетей

Как правило, при бинаризации нейронной сети значения весов и активаций ограничиваются до $+1$ и -1 (возможен также переход к значениям 0 и $+1$). Для преобразования 32-разрядного представления с плавающей точкой к бинарному используется функция бинаризации. В большинстве случаев для бинаризации используется функция знака — детерминированная функция, выдающая один и тот же результат при подаче на вход одинаковых значений входных аргументов.

$$Sign(x) = \begin{cases} +1, & \text{x} \geqslant 0 \\ -1, & \text{иначе} \end{cases}$$

Иногда используются стохастические функции бинаризации, приписывающие значения наборам аргументов с определенной вероятностью.

$$F_b(x) = \begin{cases} +1, & \text{с вероятностью } p = \sigma(x) \\ -1, & \text{с вероятностью } 1 - p, \end{cases}$$

где $\sigma(x) = \text{clip}(\frac{(x+1)}{2}, 0, 1) = \max(0, \min(1, \frac{(x+1)}{2}))$.

Несмотря на то, что применение такой функции может повысить качество работы модели, бинаризацию с ее помощью сложнее реализовать из-за того, что требуется генерация случайных бит, поэтому чаще используется функция знака.

Как и в общем случае квантизации, при бинаризации возможны два сценария: quantization-aware training (QAT), при котором 1-битные параметры и булевы операции задействованы в процессе обучения, и post-training quantization (PTQ), позволяющий напрямую квантизовать уже обученную модель даже без тонкой настройки.

В случае quantization-aware training во время прямого прохода по сети для вычисления выходов слоев вместо операции матричного умножения используется функция $XNOR$, $XNOR(A, B) = \overline{A \oplus B}$ и *popcount*, подсчитывающая количество единиц в заданном бинарном векторе.

Так как производная функции $Sign$ определена не во всех точках и почти всегда равна нулю, метод градиентного спуска не подходит для вычисления градиентов и обновления параметров при бинаризации в ходе обучения. Чтобы решить эту проблему, в [10] разработали метод STE (straight-through estimator) для приближения производной функции знака и обратного прохода по сети.

$$Approx(x) = \begin{cases} x, & \text{x} \geqslant \text{-1, x} \leqslant 1 \\ -1, & \text{x} \leqslant \text{-1} \\ +1, & \text{иначе} \end{cases}$$

$$STE(x) = \frac{\partial Approx(x)}{\partial x} = \begin{cases} 1, & \text{x} \geqslant \text{-1, x} \leqslant 1 \\ 0, & \text{иначе} \end{cases}$$

При обучении по сценарию quantization-aware training для прямого и обратного прохода по сети используются бинарные веса и активации, веса исходной модели хранятся в памяти и используются при обновлении весов перед следующим проходом по сети.

При подходе post-training quantization предобученная 32-битная модель оценивается с использованием небольшого набора калибровочных данных. Таким образом собирается статистика о распределениях весов и активаций и вычисляются калибровочные коэффициенты. Затем полученные статистики используются для квантизации модели. В ходе Post-training quantization параметры могут бинаризоваться динамически во время запуска модели с учетом входных данных или статически на основе изначальной статистики.

Подход quantization-aware training более сложен с вычислительной точки зрения, но позволяет в результате получить более точную модель,

лучше адаптированную для работы с бинарными значениями. Подход post-training quantization, исключающий алгоритм обратного распространения ошибки, проще и требует наличия только небольшого набора данных для калибровки, но бинаризованная модель оказывается менее точной.

Так как при бинаризации нейронной сети теряется существенная часть информации, точность может заметно снизиться по сравнению с изначальной моделью с 32-битными значениями. Для уменьшения ошибки бинаризации предлагались различные подходы, такие, как использование калибровочных коэффициентов при вычислении бинарных весов и активаций, оптимизация и изменение распределения весов и активаций перед бинаризацией, улучшение функции потерь с помощью добавления параметра регуляризации, дистилляция знаний [11].

3. Задачи обработки естественного языка, решенные с помощью бинарных нейронных сетей

Бинаризацию применяли для ускорения и облегчения различных языковых моделей, среди которых LSTM-сети [12], трансформерные модели, в частности, BERT [13] и LLaMA [2].

Большая часть исследований по бинаризации языковых моделей посвящена нейронным сетям, решающим задачу классификации или языкового моделирования. Задачу языкового моделирования можно сформулировать как моделирование вероятностного распределения следующего слова на основании предыдущих: $P(w_i|w_{i-1}, ..., w_0)$, где w_i — слово из словаря модели. Для оценки качества решения этой задачи используется перплексия — обратная вероятность тестовой коллекции, нормализованная по количеству слов:

$$PPL(W) = \sqrt[N]{\frac{1}{P(w_1, w_2, ..., w_N)}},$$

где W — множество слов в тестовой коллекции, N — их количество.

Первые исследования по бинаризации языковых моделей проводятся в QAT-сценарии на LSTM-сетях. В исследовании [14] адаптируют метод бинаризации, примененный ранее в задаче компьютерного зрения, для языковых моделей. Авторы [16], [20] применяют методы из теории оптимизации для подбора наилучших параметров модели, при которых достигается минимальное значение функции потерь. В [19], [22] исследуют вопрос о том, какие компоненты языковых моделей затрачивают наибольшее количество вычислительных ресурсов. Авторы показывают, что в случае LSTM-моделей это слои эмбеддингов, и предлагают техники по уменьшению потерь точности при бинаризации этих слоев, такие,

как добавление дополнительных линейных слоев и дистилляция знаний. Ниже приведенные исследования будут описаны подробнее.

Впервые бинаризация языковой модели проводится в [14]. Авторы применяют метод, ранее использованный для бинаризации свёрточных нейронных сетей Xnor-net [15], на LSTM-сети. Рассматривается 2 сценария бинаризации: бинаризация только весов модели и бинаризация весов и эмбеддингов. В первом случае в результате экспериментов было получено улучшение показателей перплексии на задаче языкового моделирования и сопоставимая точность в задаче классификации. В этом случае бинаризация выступила как регуляризатор и улучшила обобщающую способность сети. Бинаризация весов и активаций привела к переобучению и снижению точности на тестовых данных.

Авторы [16] замечают, что в прошлых исследованиях по бинаризации нейронных сетей в процессе обучения аппроксимировали матрицы весов, но не учитывали влияние бинаризации на функцию потерь. Для поиска оптимальных бинарных весов авторы используют метод Ньютона [17] — итерационный численный метод для нахождения экстремума целевой функции. Подбор бинарных весов рассматривается как оптимизационная задача, в которой требуется подобрать веса таким образом, чтобы минимизировать значение функции потерь. Матрица Гессе, участвующая в разложении функции при применении метода Ньютона, не всегда положительно определена, к тому же, ее вычисление требует больших расходов по времени и памяти, поэтому она аппроксимируется с помощью положительной диагональной матрицы. Эта матрица вычисляется с помощью моментов второго порядка, которые автоматически подсчитываются оптимизатором Adam. Результаты экспериментов на LSTM-моделях, решающих задачу языкового моделирования, показывают, что учет влияния бинаризации на функцию потерь при обучении модели позволяет уменьшить ошибку и получить более высокие результаты по сравнению с более ранними методами бинаризации [18], [15].

Авторы [19] решают проблему большого расхода памяти на слоях эмбеддингов в случае, когда размер словаря модели достаточно большой. В этом исследовании бинаризация проводится в сценарии quantization-aware training. Авторы проводят бинаризацию входного и выходного слоя эмбеддингов LSTM-сети и добавляют линейный слой после входного слоя эмбеддингов и перед выходным для улучшения точности векторных представлений. В случае бинаризации только слоев эмбеддингов авторы получили уменьшение перплексии на задаче языкового моделирования, в случае полной бинаризации модели — несущественное увеличение перплексии. Дополнительно авторы показывают, что полученные бинарные эмбеддинги не теряют информацию в сравнении с изначальными эмбеддингами с точностью представления 32 бит.

В [20] впервые при QAT-бинаризации используют метод множителей переменного направления Alternating Direction Method of Multipliers (ADMM) [21] для подбора оптимальных параметров модели, на которых значение функции потерь будет минимальным. Метод ADMM является развитием метода множителей Лагранжа и заключается в декомпозиции сложной проблемы минимизации на более простые подзадачи. При применении ADMM функция, зависящая от двух групп переменных, поочередно минимизируется то по одной, то по другой группе переменных. В [20] с помощью лагранжиана поочередно оптимизируются 2 набора параметров: веса исходной сети и значения, к которым квантизуется модель, с коэффициентом. Авторы сравнивают предложенный подход с подходом из [19] на задаче языкового моделирования и распознавания речи и показывают, что их метод позволяет ускорить сходимость сети и получить при этом более низкие значения перплексии.

В исследовании [22] бинаризацию проводят в QAT-сценарии. Эмбеддинги, на которые приходятся самые большие затраты памяти, бинаризуют в технике Product Quantization — разложением векторного пространства параметров в декартово произведение подпространств меньшей размерности и независимой квантизацией каждого подпространства. Бинаризация выполняется по методу Soft Binarization, использующему дополнительные векторы с точностью представления 32 бит для уменьшения потери информации. Метод также предполагает дистилляцию знаний на основе расстояния между распределением выходов модели-учителя ("soft labels") и модели-ученика и функции потерь этих моделей. После бинаризации модель дообучают для улучшения ее качества. Авторам удалось сжать LSTM-модель, решающую задачу языкового моделирования, в 100 раз, при этом сохранить сопоставимые с исходной моделью значения перплексии.

В исследованиях [23], [24]-[28], [31], [32], [34]-[36] проводится бинаризация трансформерных моделей, в частности, модели BERT в сценарии quantization-aware training [23], [24]-[28], больших языковых моделей [34]-[36].

Ниже приведен обзор исследований по бинаризации модели BERT. Авторы исследований частично или полностью приводят параметры модели к разрядности 1 бит, и предлагают различные способы для сохранения качества работы модели. В [23], [24], [27] предлагаются различные варианты процедуры дистилляции знаний для уменьшения ошибки квантизации, а в [26] предлагают ансамблирование нескольких бинарных моделей как более оптимальную альтернативу дистилляции. В [23], [25] используются промежуточные этапы при переходе от 32-битной модели к бинарной: инициализация весов бинарной модели от тернарной [23] и постепенное снижение разрядности модели при переходе от 32-битной модели

к бинарной [25]. В [28] бинаризация проводится на этапе предобучения модели.

В [23] впервые проводится бинаризация трансформерной модели BERT. Авторы исследуют ландшафт функции потерь и выясняют, что по мере снижения точности представления параметров модели ошибка увеличивается несущественно вплоть до 2-битной модели, в то время как ландшафт функции потерь бинарной сети оказывается более сложным, что усложняет оптимизацию и обучение модели. Для решения этой проблемы авторы обучают тернарную модель, уменьшенную по количеству параметров в 2 раза, затем с помощью оператора, отображающего тернарные веса в бинарные, инициализируют веса бинарной сети. Помимо этого, для улучшения модели авторы применяют дистилляцию знаний intermediate-layer distillation, учитывающую ошибку выхода слоя эмбеддингов, механизма внимания и линейного слоя. Результаты экспериментов на наборе задач GLUE [38] показали несущественное снижение качества работы полученной модели BinaryBERT по сравнению с 32-битными моделями и существенное улучшение показателей по сравнению с бинаризацией сети напрямую [39].

В [24] проводится полная бинаризация модели BERT. Последовательно бинаризуя различные компоненты модели, авторы замечают, что наибольшее падение точности вызывает бинаризация механизма внимания. Для решения этой проблемы они представляют структуру Bi-Attention, максимизирующую энтропию бинаризованных векторов. В структуре Bi-Attention операция матричного умножения заменяется на операцию Bitwise-Affine Matrix Multiplication, выполняющую побитовые вычисления, и устраняется операция Softmax, так как в результате применения Softmax можно получить только неотрицательные числа, и бинаризация преобразует все ее выходные значения в 1. Вместо использования Softmax авторы предлагают использовать булеву функцию, переводящую элементы attention score с низким значением в 0, и с более высоким в 1, что позволит механизму внимания выделять наиболее релевантные элементы. Для решения другой проблемы — несовпадения ожидаемого и фактического направлений градиента при оптимизации, предложили метод дистилляции Direction-Matching Distillation, учитывающий ошибку модели-ученика на матрицах запроса, ключа и значения. Эксперименты с полученной моделью, названной BiBERT, показали более высокие результаты по сравнению с другими квантизованными моделями, в том числе, BinaryBERT [23].

В [25] представляют метод Efficient Two-Stage Progressive Quantization (ETSPQ), увеличивающий степень сжатия BERT при сохранении высокого качества работы модели за счет поэтапной квантизации. Как и в [23], авторы решают проблему оптимизации бинаризованной модели с

учетом сложного ландшафта функции потерь. Авторы снижают точность представления параметров модели в 2 стадии. На первой стадии степень сжатия весов постепенно увеличивают, на каждом шаге дообучая модель на нужную задачу, затем используя параметры полученной модели для инициализации модели меньшей битности. На следующей стадии постепенно снижают разрядность активаций. В результате удалось получить большую степень сжатия и большую точность в сравнении с BinaryBERT [23] на наборе задач GLUE [38].

Авторы [26] отмечают, что используемый в прошлых бинаризованных моделях BinaryBERT [23] и BiBERT [24] метод дистилляции знаний замедляет обучение моделей, к тому же, у этих моделей существенно снижается точность и устойчивость к возмущениям во входных данных. Для решения этих проблем они предлагают использовать ансамблирование нескольких бинаризованных моделей методом AdaBoost и отказаться от дистилляции знаний. Ансамблевая модель BEBERT, построенная объединением нескольких моделей BinaryBERT или BiBERT, показала более высокие результаты на наборе задач GLUE [38], чем BiBERT [24], и сопоставимые результаты с BinaryBERT [23], и при этом обучается в 2 раза быстрее.

В [27] разрабатывают более простой способ полной бинаризации модели BERT, чем в прошлых исследованиях [23], [24]. Авторы предлагают новый подход к бинаризации: активации слоев Softmax и ReLU, принимающие только положительные значения, бинаризуются к значениям $\{0, 1\}$, в то время как активации остальных слоев, принимающие как положительные, так и отрицательные значения — к $\{-1, 1\}$, чтобы лучше сохранить свойства распределений исходных активаций. Кроме того, предлагается новый подход к процедуре дистилляции знаний: вместо того, чтобы проводить дистилляцию напрямую от модели-учителя к модели-ученику, как в BinaryBERT [23] и BiBERT [24], авторы [27] используют промежуточную модель меньшей битности, чем исходная, выступающую в роли ученика для исходной модели и в роли учителя для бинаризованной. Полученная модель BiT показала более высокие результаты, чем модели в прошлых исследованиях [23], [24].

В отличие от подходов, представленных в [23], [24] — [27], в [28] бинаризацию весов и активаций внедряют в процесс предобучения модели. Предобучение проводится на 2 стандартных для BERT задачах masked language modeling и next sentence prediction с использованием дистилляции знаний от исходной модели к бинарной. При бинаризации параметров авторы придерживаются процедуры, представленной в [27]. Авторы пытаются минимизировать ошибку бинаризации в механизме Self-Attention и вводят для этого понятие остаточных полиномов. Авторы раскладывают матрицу запроса и ключа исходной 32-битной модели в

сумму бинаризованной матрицы и остаточной матрицы, и вычисляют attention score с учетом этого разложения. Слагаемые attention score, содержащие остаточные матрицы, образуют остаточный полином, который аппроксимируется с помощью обучающихся матриц параметров. Такой подход позволяет сохранять высокие результаты, сравнимые с другими SoTA-подходами бинаризации, и при этом делает модель устойчивой к изменению гиперпараметров и позволяет дообучать ее на различные задачи.

В таблице ниже приведены результаты бинаризации трансформерной модели BERT на наборе задач GLUE.

Таблица 1. Результаты бинаризации модели BERT на наборе задач GLUE

Метод	Способ	Биты, W-A	Размер, MB	FLOPs, G	GLUE Avg
BinaryBert	QAT	1-4	16.5	1.5	82.6
		1-1	16.5	0.4	50.1
BiBERT	QAT	1-1	13.4	0.4	67.0
ETSPQ	QAT	1-4	13.4	1.5	83.2
BEBERT	QAT	1-4	33	3.0/2	82.53
		1-1	-	-	74.97
BiT	QAT	1-1	13.4	0.4	78.0
BiPFT	QAT	1-1	14.9	0.4	70.8

Исследования по бинаризации больших языковых моделей нацелены на наибольшее приближение средней разрядности параметров модели к 1 или менее бит при минимизации потерь качества. Часто для уменьшения ошибки квантизации параметры модели приводятся к смешанной разрядности, или бинаризуются только некоторые компоненты модели [29], [30], [34]. Некоторые методы бинаризации больших языковых моделей заключаются в различных стратегиях отбора наиболее значимых для обучения весов и подбора наилучшей схемы для их бинаризации [30], [32], [36], другие — в минимизации ошибки квантизации за счет использования выхода модели [33], [37]. В [31] и [34] предлагают методы для инициализации бинарных моделей, позволяющие снизить потери качества и ускоряющие сходимость модели. В [35] для сохранения лингвистической способности бинарной модели используют технологию Mixture of Experts, а в [36] для приведения модели к разрядности менее 1 бит комбинируют квантизацию и прунинг.

В [29] впервые проводится бинаризация большой языковой модели в сценарии quantization-aware training. Авторы отмечают, что прошлые

исследования по бинаризации языковых моделей проводились в основном на модели BERT [23], [24] — [27], архитектура которой существенно отличается от архитектуры больших языковых моделей. Поэтому они разрабатывают свой метод бинаризации BitNet, при котором к разрядности 1 бит приводятся только веса линейного слоя, а остальные компоненты — к разрядности 8 бит. Для ускорения обучения модели авторы используют технологию параллельных вычислений Group Quantization, при которой веса и активации модели разбиваются на несколько групп, и параметры каждой группы вычисляются отдельно. Рассмотренный метод превосходит другие SoTA-методы квантизации, такие, как SmoothQuant [6], Absmax [40] и GPTQ [41], по уменьшению затрат памяти и энергии, при этом показывает сопоставимое качество на тестовых данных.

Как и в [29], в [30] отмечают невозможность адаптировать существующие методы бинаризации для больших языковых моделей, и предлагают новый подход Partially-Binarized LLM (PB-LLM), сохраняющий лингвистическую способность языковых моделей. Подход основан на отборе небольшого количества наиболее значимых для обучения весов и сохранении их в высокой разрядности, и бинаризации остальных весов. Метод применяется как для бинаризации в процессе обучения, так и для бинаризации уже обученной модели. В первом случае авторы замораживают наиболее значимые веса, отобранные по величине, и затем обучают бинаризованную модель. Во втором случае авторы обобщают метод квантизации Gptq [41] для бинаризации: итеративно бинаризуют незначимые веса и квантизуют к более высокой разрядности значимые, затем применяют к оставшимся весам компенсирующий коэффициент для уменьшения ошибки квантизации. Эксперименты на модели LLaMA [2] показали результаты, сравнимые с другими SoTA-методами квантизации SmoothQuant [6], Llm-qat [42] и RTN [43] в случае сохранения 30% весов в высокой разрядности при бинаризации после обучения, и более высокие результаты, чем другие методы квантизации в случае бинаризации в ходе обучения, обеспечивая при этом быструю сходимость сети.

Авторы [31] отмечают, что смешение разных разрядностей в PB-LLM [30] усложняет применение подхода и ограничивает экономию памяти. Они предлагают подход QAT-бинаризации Dual-Binarization (DB-LLM), при котором достигается удобный для оптимизации ландшафт функции потерь и используются булевы операции, оптимизирующие затраты вычислительных ресурсов. Авторы пользуются тем, что ландшафт функции потерь 2-битной модели более плоский и удобный для оптимизации, чем у бинарной, и раскладывают веса 2-битной модели в сумму бинарных весов, домноженных на калибровочный коэффициент. Полученные веса используются для инициализации бинарной модели, а калибровочные коэффициенты дополнительно настраиваются на этапе тонкой настрой-

ки. Другая проблема, замеченная авторами [31], заключается в том, что квантизованная модель по статистике более склонна предсказывать часто встречающиеся классы. Чтобы ее решить, предлагается подход Deviation-aware Distillation, уравновешивающий примеры из частых и редких классов, в котором энтропия модели-ученика и энтропия модели-учителя служат мерой неуверенности модели при предсказании классов. Бинаризация модели LLaMA [2] способом DB-LLM превосходит SoTA-методы квантизации, такие как Gptq [41], RTN [43] и PB-LLM [30] и позволяет добиться меньшей вычислительной сложности.

Еще одна попытка усовершенствовать метод PB-LLM в случае PTQ-бинаризации предпринимается в [32]. Авторы отмечают высокие затраты памяти метода PB-LLM, требующего хранения более 30% весов в высокой разрядности. Идея предложенного ими подхода BiLLM состоит в отборе значимых и незначимых для обучения весов и их бинаризации по двум различным схемам. Так как распределение значимых весов обладает высокой дисперсией, стандартные схемы post-training бинаризации для них не подходят. Предлагается новая схема бинаризации, по которой значимые веса бинаризуются рекурсивно: для бинаризованной матрицы параметров вычисляется остаточная матрица, и тот же процесс бинаризации применяется уже к ней. Авторы замечают, что оставшиеся незначимые веса имеют нормальное распределение, и исходя из этого, подбирают порог, разбивающий веса на две категории: сконцентрированные и разреженные, и вычисляют ошибку бинаризации для незначимых весов как сумму ошибки на разреженном и на сконцентрированном участке. Эксперименты на моделях семейства LLaMA [2] и OPT [44] показывают существенное улучшение показателей перплексии по сравнению с другими SoTA-методами квантизации Gptq [41], RTN [43] и PB-LLM [30], при этом наибольшее приближение средней разрядности весов модели к 1 бит.

Другой подход к PTQ-бинаризации Output-adaptive Calibration (OAC), учитывающий выход модели при квантизации, описан в [33]. Вместо того, чтобы вычислять ошибку бинаризации между выходом квантизованного и исходного слоя, авторы минимизируют расхождение между функцией потерь на выходе модели до и после квантизации. При нахождении этого расхождения метод OAC использует вторую производную перекрёстной энтропии для вычисления гессиана. Вычисленный гессиан используется для обновления весов и определения их значимости. Предложенный подход показывает лучшие результаты на моделях LLaMA [2] и OPT [44], чем другие подходы квантизации OPTQ [45], QuIP [46], SpQR [47] и BiLLM [32], не учитывающие выход модели при минимизации ошибки бинаризации.

Авторы подхода OneBit [34], предполагающего quantization-aware training, бинаризуют линейный слой по методу, представленному в BitNet

[29], но дополняют его двумя векторами разрядности 16 бит g и h, на которые покомпонентно домножаются столбцы матрицы активаций и матрицы бинарных весов, соответственно. Для инициализации бинарной модели авторы представляют метод Sign-Value-Independent Decomposition, при котором матрица весов из линейного слоя раскладывается в покомпонентное произведение матрицы знака и матрицы значений, а матрица значений затем раскладывается в произведение двух векторов, выполняющих роль векторов g и h при первом запуске. Для повышения качества работы модели применяется дистилляция знаний. Бинаризованная таким методом модель LLaMA [2] более стабильна к изменению гиперпараметров и показывает результаты, сравнимые с 16-битными моделями.

Авторы [35] решают проблему низкой точности бинаризованных моделей и представляют метод QAT-бинаризации Mixture of Scales (BinaryMoS), затрачивающий небольшое дополнительное количество памяти, но существенно улучшающий лингвистические способности модели. Метод вдохновлен структурой Mixture of Experts [48], дублирующей слои модели и выбирающей подходящий для данной задачи слой (эксперт) среди дубликатов во время запуска модели на основе коэффициентов, приписанных маршрутизатором. В качестве экспертов в случае BinaryMoS выступают вектора калибровочных коэффициентов, а маршрутизатор, отбирающий наиболее подходящие из них в зависимости от входного токена, представлен линейным слоем с функцией активации Softmax. Пользуясь тем, что коэффициенты задействованы только в линейных слоях, BinaryMoS динамически генерирует инструкции о том, как линейно комбинировать вектора коэффициентов, что позволяет не ограничиваться фиксированным числом экспертов. Как и в прошлых подходах, авторы применяют дистилляцию знаний для улучшения работы модели. Подход BinaryMoS показал более высокие результаты на моделях LLaMA [2] и OPT [44], чем прошлые подходы PB-LLM [30], OneBit [34] и BiLLM [32], при этом сохраняя низкие затраты по памяти.

Подход post-training quantization, названный STructured Binarization for LLMs (STBLLM) [36], сочетает в себе сразу две техники сжатия модели — бинаризацию и прунинг. Для приведения модели к средней разрядности менее 1 бит авторы проводят прунинг весов предобученной модели по методу N:M Sparsity [49], который кодирует N последовательных ненулевых элементов матрицы весов с помощью чисел в M-битном представлении, затем бинаризуют модель. Как и в прошлых исследованиях, веса модели разделяются на значимые и незначимые. Авторы представляют новую метрику для отбора значимых весов на основе их величины — Standardized Importance, не использующую гессиан и упрощающую вычисления. Значимые веса бинаризуются способом, представленным в BiLLM [32], а незначимые, как и в BiLLM, разбиваются на группы на основе их рас-

пределения. Метод STBLLM позволяет получить лучшие результаты на моделях LLaMA [2], OPT [44] и Mistral [50], чем BiLLM, и показывает высокий потенциал дальнейшего сжатия моделей до разрядности менее 1 бит.

Метод обучения бинаризованной большой языковой модели с нуля в сценарии quantization-aware training предлагается в [37]. Авторы бинаризуют только линейные слои модели с использованием калибровочных коэффициентов, а для уменьшения ошибки бинаризации используют авторегрессионную дистилляцию знаний, при которой на каждом шаге предсказания следующего токена вычисляется перекрестная энтропия между распределением вероятностей выходного токена 32-битной модели-учителя и бинарной модели-ученика. Эксперименты показали, что использование такой функции дает достаточно высокие результаты, поэтому другие слагаемые в функцию потерь не включаются. Предложенный подход бинаризации Fully BInarized Large Language Model (FBI-LLM), примененный к моделям LLaMA [2] и OPT [44], показывает более высокие результаты в большинстве экспериментов, чем другие SoTA-методы бинаризации BiLLM [32], OneBit [34], BitNet [29].

Ниже приведена таблица с результатами бинаризации больших языковых моделей.

Таблица 2. Результаты бинаризации больших языковых моделей

Метод	Способ	Модель	Биты	Корпус	Метрика	Значение
BitNet	QAT	Transformer	-	HellaSwag	Acc	38.9
				Winogrande	Acc	51.4
PB-LLM	QAT	LLaMA-1-7B	1.70	WikiText2	PPL	20.61
				C4	PPL	47.09
PB-LLM	PTQ	LLaMA-1-7B	1.70	WikiText2	PPL	102.36
		OPT-1.3B	1.70	WikiText2	PPL	265.52
		OPT-13B	1.70	WikiText2	PPL	81.92
DB-LLM	QAT	LLaMA-1-7B	-	WikiText2	PPL	7.59
				C4	PPL	9.74
BiLLM	PTQ	LLaMA-1-7B	1.08	WikiText2	PPL	35.04
			1.08	C4	PPL	39.6
		LLaMA2-7B	1.08	WikiText2	PPL	32.5
			1.08	C4	PPL	40.5
		OPT-1.3B	1.11	WikiText2	PPL	35.4
			1.11	C4	PPL	43.2
		OPT-1.3B	0.55	Wikitext2	PPL	106.99
		Mistral	0.55	Wikitext2	PPL	189.73
OAC	PTQ	LLaMA-1-7B	1.09	WikiText2	PPL	17.79
			1.09	C4	PPL	19.82
		OPT-13B	2.10	WikiText2	PPL	11.75
			2.10	C4	PPL	13.25
OneBit	QAT	LLaMA-1-7B	-	WikiText2	PPL	10.19
				C4	PPL	11.40
		LLaMA-2-7B	-	WikiText2	PPL	9.7
				C4	PPL	11.1
BinaryMoS	QAT	LLaMA-1-7B	1.0	WikiText2	PPL	7.97
			1.0	C4	PPL	9.72
		OPT-1.3B	1.0	WikiText2	PPL	18.45
			1.0	C4	PPL	18.83
STBLLM	PTQ	LLaMA-1-7B	0.55	Wikitext2	PPL	31.72
		OPT-1.3B	0.55	Wikitext2	PPL	45.11
		Mistral	0.55	Wikitext2	PPL	70.14
FBI-LLM	QAT	LLaMA-2-7B	1.01	Wikitext2	PPL	5.7
			1.01	C4	PPL	7.3
			1.01	HellaSwag	Acc	57.7
			1.01	Winogrande	Acc	58.9
		OPT-1.3B	1.01	Wikitext2	PPL	12.6
			1.01	C4	PPL	13.8

4. Заключение

Были рассмотрены основные подходы к бинаризации языковых моделей, в том числе, исследования по бинаризации рекуррентных сетей [14] — [22], модели BERT [23], [24] — [28], и последние исследования по бинаризации больших языковых моделей [31], [32], [34]. Для повышения точности и уменьшения средней разрядности языковых моделей предлагались различные математические и технические решения, но можно выделить несколько тенденций.

Большинство исследований по бинаризации языковых моделей проводятся в QAT-сценарии, так как такой подход позволяет получить меньшее падение точности моделей и лучше адаптировать их для работы с бинарными параметрами. Тем не менее, направление post-training binarization кажется перспективным для будущих исследований — предлагаются различные техники по бинаризации параметров, и эти подходы обобщаются для различных архитектур, таких, как LLaMA, OPT, Mistral [32].

Несмотря на популярность подхода quantization-aware training, некоторые проблемы остаются нерешенными. В частности, бинаризация не способна давать высокие результаты без применения дополнительных методов оптимизации, таких, как дистилляция знаний [23], ансамблирование [26] или дополнительные стадии обучения [25], которые затрачивают дополнительные вычислительные ресурсы и время.

Одним из популярных направлений исследований по бинаризации является сохранение информации бинарных представлений при прямом и обратном проходе по сети. Для этого предлагались такие методы, как максимизация энтропии бинаризованных векторов [24], метод дистилляции знаний, учитывающий направление градиента [24], заморозка наиболее значимых весов в высокой разрядности [30].

Еще одним перспективным направлением исследований является снижение средней разрядности бинаризованных моделей. Фактически, бинаризованные модели часто имеют среднюю точность представления более 1 бит, так как для уменьшения потерь в качестве требуется сохранение части параметров в более высокой разрядности, чем 1 бит. Тем не менее, в недавних исследованиях достигается все более и более низкая разрядность параметров [36].

Бинаризация является мощной техникой оптимизации языковых моделей, которая может позволить внедрить большие языковые модели на пользовательские устройства с ограниченным количеством памяти. В качестве направлений будущих исследований можно выделить комбинирование различных техник оптимизации для достижения наибольшего сжатия моделей, таких, как бинаризация и прунинг; уменьшение падения точности по сравнению с исходными 32-битными моделями и разработку

универсальной схемы бинаризации для различных языковых задач, таких, как распознавание речи, классификация и языковое моделирование.

Список литературы

[1] OpenAI, "GPT-4 Technical Report", *arXiv preprint arXiv:2303.08774*, 2023, 100 pp., arXiv: arXiv:2303.08774

[2] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, Guillaume Lample, "LLaMA: Open and Efficient Foundation Language Models", *arXiv preprint arXiv:2302.13971*, 2023, 27 pp., arXiv: arXiv:2302.13971

[3] Minjia Zhang, Yuxiong He, "Accelerating training of transformer-based language models with progressive layer dropping", *Advances in Neural Information Processing Systems*, **33**, Neural Information Processing Systems Foundation, 2020, 14011–14023

[4] Yujie Zeng, Wenlong He, Ihor Vasyltsov, Jiali Pang, Lin Chen, "Acceleration of large transformer model training by sensitivity-based layer dropping", *Proceedings of the AAAI Conference on Artificial Intelligence*, **37**, Association for the Advancement of Artificial Intelligence (AAAI), 2023, 11156–11163

[5] Zhewei Yao, Reza Y. Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, Yuxiong He, "Zeroquant: Efficient and affordable post-training quantization for large-scale transformers", *Advances in Neural Information Processing Systems*, **35**, Neural Information Processing Systems Foundation, 2022, 27168–27183

[6] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, Song Han, "Smoothquant: Accurate and efficient post-training quantization for large language models", *Proceedings of the 40th International Conference on Machine Learning*, **202**, Proceedings of Machine Learning Research (PMLR), 2023, 38087–38099

[7] Yann LeCun, John S. Denker, Sara A. Solla, "Optimal brain damage", *Proceedings of the 3rd International Conference on Neural Information Processing Systems*, **2**, MIT Press, 1989, 598–605

[8] Babak Hassibi, David G. Stork, "Second order derivatives for network pruning: Optimal brain surgeon", *Proceedings of the 6th International*

Conference on Neural Information Processing Systems (NIPS'92), **5**, Morgan Kaufmann Publishers Inc., 1992, 164–171

[9] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, "Distilling the Knowledge in a Neural Network", *arXiv preprint arXiv:1503.02531*, 2015, 6 pp., arXiv: arXiv:1503.02531

[10] Yoshua Bengio, Nicholas Léonard, Aaron Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation", *arXiv preprint arXiv:1308.3432*, 2013, 12 pp., arXiv: arXiv:1308.3432

[11] Chunyu Yuan, Sos S. Agaian, "A comprehensive review of binary neural network", *Artificial Intelligence Review*, **56**:11 (2023), 12949–13013

[12] Sepp Hochreiter, Jürgen Schmidhuber, "Long short-term memory", *Neural Computation*, **9**:8 (1997), 1735–1780

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, **1**, Association for Computational Linguistics, 2019, 4171–4186

[14] Weiyi Zheng, Yina Tang, "Binarized neural networks for language modeling", *Technical Report CS224d, Stanford University*, 2016, 9 pp.

[15] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, Ali Farhadi, "XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks", *Proceedings of the 14th European Conference on Computer Vision (ECCV 2016)*, **9908**, Springer, 2016, 525–542

[16] Lu Hou, Quanming Yao, James T. Kwok, "Loss-aware Binarization of Deep Networks", *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*, International Conference on Learning Representations (ICLR), 2017, 9 pp.

[17] Jason D. Lee, Yuekai Sun, Michael A. Saunders, "Proximal Newton-type methods for minimizing composite functions", *SIAM Journal on Optimization*, **24**:3 (2014), 1420–1443

[18] Matthieu Courbariaux, Yoshua Bengio, Jean-Pierre David, "BinaryConnect: Training deep neural networks with binary weights during propagations", *Advances in Neural Information Processing Systems (NeurIPS)*, **28**, Curran Associates, Inc., 2015, 3123–3131

[19] Xuan Liu, Di Cao, Kai Yu, "Binarized LSTM Language Model", *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, Association for Computational Linguistics, 2018, 2113–2121

[20] Junhao Xu, Xie Chen, Shoukang Hu, Jianwei Yu, Xunying Liu, Helen Meng, "Low-bit Quantization of Recurrent Neural Network Language Models Using Alternating Direction Methods of Multipliers", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE Press, 2020, 7939–7943

[21] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers", *Foundations and Trends in Machine Learning*, **3**:1 (2011), 1–122

[22] Kai Yu, Rao Ma, Kaiyu Shi, Qi Liu, "Neural Network Language Model Compression With Product Quantization and Soft Binarization", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, **28**:10 (2020), 2438–2449

[23] Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jing Jin, Xin Jiang, Qun Liu, Michael Lyu, Irwin King, "BinaryBERT: Pushing the Limit of BERT Quantization", *arXiv preprint arXiv:2012.15701*, 2020, 13 pp., arXiv: arXiv:2012.15701

[24] Haotong Qin, Yifu Ding, Mingyuan Zhang, Qinghua Yan, Aishan Liu, Qingqing Dang, Ziwei Liu, Xianglong Liu, "BiBERT: Accurate Fully Binarized BERT", *Proceedings of the International Conference on Learning Representations (ICLR)*, International Conference on Learning Representations, 2022, 12 pp.

[25] Phuc H. C. Le, *Towards Accurate Low-Bitwidth BERT*, McGill University, Montreal, Canada, 2023, 120 pp.

[26] Jie Tian, Chen Fang, Hui Wang, Zhiyuan Wang, "BEBERT: Efficient and Robust Binary Ensemble BERT", *Proceedings of the 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE Press, 2023, 1–5

[27] Zechun Liu, Barlas Oğuz, Aasish Pappu, Lin Xiao, Scott Yih, Meng Li, Raghuraman Krishnamoorthi, Yashar Mehdad, "BiT: Robustly Binarized Multi-Distilled Transformer", *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS '22)*, **35**, Curran Associates Inc., 2022, 14303–14316

[28] Xingrun Xing, Li Du, Xinyuan Wang, Xianlin Zeng, Yequan Wang, Zheng Zhang, Jiajun Zhang, "BiPFT: Binary Pre-trained Foundation Transformer with Low-Rank Estimation of Binarization Residual Polynomials", *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence (AAAI'24/IAAI'24/EAAI'24)*, **38**, AAAI Press, 2024, 16094–16102

[29] Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, Furu Wei, "BitNet: Scaling 1-bit Transformers for Large Language Models", *arXiv preprint arXiv:2310.11453*, 2023, 15 pp., arXiv: arXiv:2310.11453

[30] Yuzhang Shang, Zhihang Yuan, Qiang Wu, Zhen Dong, "PB-LLM: Partially Binarized Large Language Models", *arXiv preprint arXiv:2310.00034*, 2023, 14 pp., arXiv: arXiv:2310.00034

[31] Hong Chen, Chengtao Lv, Liang Ding, Haotong Qin, Xiabin Zhou, Yifu Ding, Xuebo Liu, Min Zhang, Jinyang Guo, Xianglong Liu, Dacheng Tao, "DB-LLM: Accurate Dual-Binarization for Efficient LLMs", *Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024*, **1**, Association for Computational Linguistics, 2024, 8719–8730

[32] Wei Huang, Yangdong Liu, Haotong Qin, Ying Li, Shiming Zhang, Xianglong Liu, Michele Magno, Xiaojuan Qi, "BiLLM: Pushing the Limit of Post-Training Quantization for LLMs", *Proceedings of the 41st International Conference on Machine Learning (ICML)*, **202**, JMLR.org, 2024, 12950–12969

[33] Ali Edalati, Alireza Ghaffari, Masoud Asgharian, Lu Hou, Boxing Chen, Vahid Partovi Nia, "OAC: Output-Adaptive Calibration for Accurate Post-Training Quantization", *arXiv preprint arXiv:2405.15025*, 2024, 13 pp., arXiv: arXiv:2405.15025

[34] Yuzhuang Xu, Xu Han, Zonghan Yang, Shuo Wang, Qingfu Zhu, Zhiyuan Liu, Weidong Liu, Wanxiang Che, "OneBit: Towards Extremely Low-Bit Large Language Models", *Advances in Neural Information Processing Systems (NeurIPS)*, **37** (2024), 1–14

[35] Dongwon Jo, Taesu Kim, Yulhwa Kim, Jae-Joon Kim, "Mixture of Scales: Memory-Efficient Token-Adaptive Binarization for Large Language Models", *arXiv preprint arXiv:2406.12311*, 2024, 11 pp., arXiv: arXiv:2406.12311

[36] Peijie Dong, Lujun Li, Dayou Du, Yuhan Chen, Zhenheng Tang, Qiang Wang, Wei Xue, Wenhan Luo, Qifeng Liu, Yike Guo, Xiaowen Chu, "STBLLM: Breaking the 1-Bit Barrier with Structured Binary LLMs", *arXiv preprint arXiv:2408.01803*, 2024, 23 pp., arXiv: arXiv:2408.01803

[37] Liqun Ma, Mingjie Sun, Zhiqiang Shen, "FBI-LLM: Scaling Up Fully Binarized LLMs from Scratch via Autoregressive Distillation", *arXiv preprint arXiv:2407.07093*, 2024, 18 pp., arXiv: arXiv:2407.07093

[38] Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, Samuel Bowman, "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding", *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, **57**, Association for Computational Linguistics, 2018, 353–355

[39] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, Yoshua Bengio, "Binarized Neural Networks", *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16)*, **29**, Curran Associates Inc., 2016, 4114–4122

[40] Tim Dettmers, Mike Lewis, Younes Belkada, Luke Zettlemoyer, "GPT3.int8(): 8-bit Matrix Multiplication for Transformers at Scale", *Advances in Neural Information Processing Systems*, **35**, Curran Associates, Inc., 2022, 30318–30332

[41] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, Dan Alistarh, "GPTQ: Accurate Post-Training Quantization for Generative Pre-Trained Transformers", *arXiv preprint arXiv:2210.17323*, 2023, 9 pp., arXiv: arXiv:2210.17323

[42] Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, Vikas Chandra, "LLM-QAT: Data-Free Quantization Aware Training for Large Language Models", *Findings of the Association for Computational Linguistics: ACL 2024*, 2024, 467–484

[43] Yuhang Li, Xin Dong, Sai Zhang, Haoli Bai, Yuanpeng Chen, Wei Wang, "RTN: Reparameterized Ternary Network", *Proceedings of the AAAI Conference on Artificial Intelligence*, **34**, AAAI Press, 2020, 4780–4787

[44] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Luke Zettlemoyer, "OPT: Open Pre-trained Transformer Language Models", *arXiv preprint arXiv:2205.01068*, 2022, 40 pp., arXiv: arXiv:2205.01068

[45] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, Dan Alistarh, "OPTQ: Accurate quantization for generative pre-trained transformers", *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, International Conference on Learning Representations, 2023, 11 pp.

[46] Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, Christopher De Sa, "QuIP: 2-Bit Quantization of Large Language Models with Guarantees", *Advances in Neural Information Processing Systems*, **36** (2023), 37371–37382

[47] Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, Dan Alistarh, "SpQR: A Sparse-Quantized Representation for Near-Lossless LLM Weight Compression", *arXiv preprint arXiv:2306.03078*, 2023, 15 pp., arXiv: arXiv:2306.03078

[48] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, Jeff Dean, "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer", *arXiv preprint arXiv:1701.06538*, 2017, 13 pp., arXiv: arXiv:1701.06538

[49] Asit Mishra, Jorge A. Latorre, Jeff Pool, Darko Stosic, Dusan Stosic, Ganesh Venkatesh, Chong Yu, Paulius Micikevicius, "Accelerating Sparse Deep Neural Networks", *arXiv preprint arXiv:2104.08378*, 2021, 9 pp., arXiv: arXiv:2104.08378

[50] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, William El Sayed, "Mistral 7B", *arXiv preprint arXiv:2310.06825*, 2023, 33 pp., arXiv: arXiv:2310.06825

Binarization of language models

Davydova D.N.

Large language models are widely used in the field of natural language processing. However, despite their high efficiency, the application of large language models becomes difficult due to their high computational and memory costs.

One of the ways to solve this problem is neural network quantization, that is, converting the weights and activations of the network to a representation with lower bit-width. A special case of quantization is binarization, which is the compression of network parameters to a bit-width of 1 bit.

In this paper, the structure of binary neural networks is considered, an overview of current methods of language model binarization is provided, and the results obtained are described.

Keywords: natural language processing, binary neural networks, binarization, quantization, large language models.

References

[1] OpenAI, "GPT-4 Technical Report", *arXiv preprint arXiv:2303.08774*, 2023, 100 pp., arXiv: arXiv:2303.08774

[2] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, Guillaume Lample, "LLaMA: Open and Efficient Foundation Language Models", *arXiv preprint arXiv:2302.13971*, 2023, 27 pp., arXiv: arXiv:2302.13971

[3] Minjia Zhang, Yuxiong He, "Accelerating training of transformer-based language models with progressive layer dropping", *Advances in Neural Information Processing Systems*, **33**, Neural Information Processing Systems Foundation, 2020, 14011–14023

[4] Yujie Zeng, Wenlong He, Ihor Vasyltsov, Jiali Pang, Lin Chen, "Acceleration of large transformer model training by sensitivity-based layer dropping", *Proceedings of the AAAI Conference on Artificial Intelligence*, **37**, Association for the Advancement of Artificial Intelligence (AAAI), 2023, 11156–11163

[5] Zhewei Yao, Reza Y. Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, Yuxiong He, "Zeroquant: Efficient and affordable post-training quantization for large-scale transformers", *Advances in Neural Information Processing Systems*, **35**, Neural Information Processing Systems Foundation, 2022, 27168–27183

[6] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, Song Han, "Smoothquant: Accurate and efficient post-training quantization for large language models", *Proceedings of the 40th International Conference on Machine Learning*, **202**, Proceedings of Machine Learning Research (PMLR), 2023, 38087–38099

[7] Yann LeCun, John S. Denker, Sara A. Solla, "Optimal brain damage", *Proceedings of the 3rd International Conference on Neural Information Processing Systems*, **2**, MIT Press, 1989, 598–605

[8] Babak Hassibi, David G. Stork, "Second order derivatives for network pruning: Optimal brain surgeon", *Proceedings of the 6th International Conference on Neural Information Processing Systems (NIPS'92)*, **5**, Morgan Kaufmann Publishers Inc., 1992, 164–171

[9] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, "Distilling the Knowledge in a Neural Network", *arXiv preprint arXiv:1503.02531*, 2015, 6 pp., arXiv: arXiv:1503.02531

[10] Yoshua Bengio, Nicholas Léonard, Aaron Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation", *arXiv preprint arXiv:1308.3432*, 2013, 12 pp., arXiv: arXiv:1308.3432

[11] Chunyu Yuan, Sos S. Agaian, "A comprehensive review of binary neural network", *Artificial Intelligence Review*, **56**:11 (2023), 12949–13013

[12] Sepp Hochreiter, Jürgen Schmidhuber, "Long short-term memory", *Neural Computation*, **9**:8 (1997), 1735–1780

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, **1**, Association for Computational Linguistics, 2019, 4171–4186

[14] Weiyi Zheng, Yina Tang, "Binarized neural networks for language modeling", *Technical Report CS224d, Stanford University*, 2016, 9 pp.

[15] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, Ali Farhadi, "XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks", *Proceedings of the 14th European Conference on Computer Vision (ECCV 2016)*, **9908**, Springer, 2016, 525–542

[16] Lu Hou, Quanming Yao, James T. Kwok, "Loss-aware Binarization of Deep Networks", *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*, International Conference on Learning Representations (ICLR), 2017, 9 pp.

[17] Jason D. Lee, Yuekai Sun, Michael A. Saunders, "Proximal Newton-type methods for minimizing composite functions", *SIAM Journal on Optimization*, **24**:3 (2014), 1420–1443

[18] Matthieu Courbariaux, Yoshua Bengio, Jean-Pierre David, "BinaryConnect: Training deep neural networks with binary weights during propagations", *Advances in Neural Information Processing Systems (NeurIPS)*, **28**, Curran Associates, Inc., 2015, 3123–3131

[19] Xuan Liu, Di Cao, Kai Yu, "Binarized LSTM Language Model", *Proceedings of the 2018 Conference of the North American Chapter*

of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, 2018, 2113–2121

[20] Junhao Xu, Xie Chen, Shoukang Hu, Jianwei Yu, Xunying Liu, Helen Meng, "Low-bit Quantization of Recurrent Neural Network Language Models Using Alternating Direction Methods of Multipliers", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE Press, 2020, 7939–7943

[21] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers", *Foundations and Trends in Machine Learning*, **3**:1 (2011), 1–122

[22] Kai Yu, Rao Ma, Kaiyu Shi, Qi Liu, "Neural Network Language Model Compression With Product Quantization and Soft Binarization", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, **28**:10 (2020), 2438–2449

[23] Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jing Jin, Xin Jiang, Qun Liu, Michael Lyu, Irwin King, "BinaryBERT: Pushing the Limit of BERT Quantization", *arXiv preprint arXiv:2012.15701*, 2020, 13 pp., arXiv: arXiv:2012.15701

[24] Haotong Qin, Yifu Ding, Mingyuan Zhang, Qinghua Yan, Aishan Liu, Qingqing Dang, Ziwei Liu, Xianglong Liu, "BiBERT: Accurate Fully Binarized BERT", *Proceedings of the International Conference on Learning Representations (ICLR)*, International Conference on Learning Representations, 2022, 12 pp.

[25] Phuc H. C. Le, *Towards Accurate Low-Bitwidth BERT*, McGill University, Montreal, Canada, 2023, 120 pp.

[26] Jie Tian, Chen Fang, Hui Wang, Zhiyuan Wang, "BEBERT: Efficient and Robust Binary Ensemble BERT", *Proceedings of the 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE Press, 2023, 1–5

[27] Zechun Liu, Barlas Oğuz, Aasish Pappu, Lin Xiao, Scott Yih, Meng Li, Raghuraman Krishnamoorthi, Yashar Mehdad, "BiT: Robustly Binarized Multi-Distilled Transformer", *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS '22)*, **35**, Curran Associates Inc., 2022, 14303–14316

[28] Xingrun Xing, Li Du, Xinyuan Wang, Xianlin Zeng, Yequan Wang, Zheng Zhang, Jiajun Zhang, "BiPFT: Binary Pre-trained Foundation Transformer with Low-Rank Estimation of Binarization Residual Polynomials", *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence (AAAI'24/IAAI'24/EAAI'24)*, **38**, AAAI Press, 2024, 16094–16102

[29] Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, Furu Wei, "BitNet: Scaling 1-bit Transformers for Large Language Models", *arXiv preprint arXiv:2310.11453*, 2023, 15 pp., arXiv: arXiv:2310.11453

[30] Yuzhang Shang, Zhihang Yuan, Qiang Wu, Zhen Dong, "PB-LLM: Partially Binarized Large Language Models", *arXiv preprint arXiv:2310.00034*, 2023, 14 pp., arXiv: arXiv:2310.00034

[31] Hong Chen, Chengtao Lv, Liang Ding, Haotong Qin, Xiabin Zhou, Yifu Ding, Xuebo Liu, Min Zhang, Jinyang Guo, Xianglong Liu, Dacheng Tao, "DB-LLM: Accurate Dual-Binarization for Efficient LLMs", *Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024*, **1**, Association for Computational Linguistics, 2024, 8719–8730

[32] Wei Huang, Yangdong Liu, Haotong Qin, Ying Li, Shiming Zhang, Xianglong Liu, Michele Magno, Xiaojuan Qi, "BiLLM: Pushing the Limit of Post-Training Quantization for LLMs", *Proceedings of the 41st International Conference on Machine Learning (ICML)*, **202**, JMLR.org, 2024, 12950–12969

[33] Ali Edalati, Alireza Ghaffari, Masoud Asgharian, Lu Hou, Boxing Chen, Vahid Partovi Nia, "OAC: Output-Adaptive Calibration for Accurate Post-Training Quantization", *arXiv preprint arXiv:2405.15025*, 2024, 13 pp., arXiv: arXiv:2405.15025

[34] Yuzhuang Xu, Xu Han, Zonghan Yang, Shuo Wang, Qingfu Zhu, Zhiyuan Liu, Weidong Liu, Wanxiang Che, "OneBit: Towards Extremely Low-Bit Large Language Models", *Advances in Neural Information Processing Systems (NeurIPS)*, **37** (2024), 1–14

[35] Dongwon Jo, Taesu Kim, Yulhwa Kim, Jae-Joon Kim, "Mixture of Scales: Memory-Efficient Token-Adaptive Binarization for Large Language Models", *arXiv preprint arXiv:2406.12311*, 2024, 11 pp., arXiv: arXiv:2406.12311

[36] Peijie Dong, Lujun Li, Dayou Du, Yuhan Chen, Zhenheng Tang, Qiang Wang, Wei Xue, Wenhan Luo, Qifeng Liu, Yike Guo, Xiaowen Chu, "STBLLM: Breaking the 1-Bit Barrier with Structured Binary LLMs", *arXiv preprint arXiv:2408.01803*, 2024, 23 pp., arXiv: arXiv:2408.01803

[37] Liqun Ma, Mingjie Sun, Zhiqiang Shen, "FBI-LLM: Scaling Up Fully Binarized LLMs from Scratch via Autoregressive Distillation", *arXiv preprint arXiv:2407.07093*, 2024, 18 pp., arXiv: arXiv:2407.07093

[38] Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, Samuel Bowman, "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding", *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, **57**, Association for Computational Linguistics, 2018, 353–355

[39] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, Yoshua Bengio, "Binarized Neural Networks", *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16)*, **29**, Curran Associates Inc., 2016, 4114–4122

[40] Tim Dettmers, Mike Lewis, Younes Belkada, Luke Zettlemoyer, "GPT3.int8(): 8-bit Matrix Multiplication for Transformers at Scale", *Advances in Neural Information Processing Systems*, **35**, Curran Associates, Inc., 2022, 30318–30332

[41] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, Dan Alistarh, "GPTQ: Accurate Post-Training Quantization for Generative Pre-Trained Transformers", *arXiv preprint arXiv:2210.17323*, 2023, 9 pp., arXiv: arXiv:2210.17323

[42] Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, Vikas Chandra, "LLM-QAT: Data-Free Quantization Aware Training for Large Language Models", *Findings of the Association for Computational Linguistics: ACL 2024*, 2024, 467–484

[43] Yuhang Li, Xin Dong, Sai Zhang, Haoli Bai, Yuanpeng Chen, Wei Wang, "RTN: Reparameterized Ternary Network", *Proceedings of the AAAI Conference on Artificial Intelligence*, **34**, AAAI Press, 2020, 4780–4787

[44] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Luke Zettlemoyer, "OPT: Open Pre-trained Transformer Language Models", *arXiv preprint arXiv:2205.01068*, 2022, 40 pp., arXiv: arXiv:2205.01068

[45] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, Dan Alistarh, "OPTQ: Accurate quantization for generative pre-trained transformers", *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, International Conference on Learning Representations, 2023, 11 pp.

[46] Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, Christopher De Sa, "QuIP: 2-Bit Quantization of Large Language Models with Guarantees", *Advances in Neural Information Processing Systems*, **36** (2023), 37371–37382

[47] Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, Dan Alistarh, "SpQR: A Sparse-Quantized Representation for Near-Lossless LLM Weight Compression", *arXiv preprint arXiv:2306.03078*, 2023, 15 pp., arXiv: arXiv:2306.03078

[48] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, Jeff Dean, "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer", *arXiv preprint arXiv:1701.06538*, 2017, 13 pp., arXiv: arXiv:1701.06538

[49] Asit Mishra, Jorge A. Latorre, Jeff Pool, Darko Stosic, Dusan Stosic, Ganesh Venkatesh, Chong Yu, Paulius Micikevicius, "Accelerating Sparse Deep Neural Networks", *arXiv preprint arXiv:2104.08378*, 2021, 9 pp., arXiv: arXiv:2104.08378

[50] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, William El Sayed, "Mistral 7B", *arXiv preprint arXiv:2310.06825*, 2023, 33 pp., arXiv: arXiv:2310.06825

О трех начальных приближениях к формальному определению визуального образа в произвольной визуальной среде

В. Н. Козлов[1]

Распознавание визуальных образов — одна из центральных задач для интеллектуальных систем. Продвижению здесь математических, теоремных методов исследования в немалой степени мешает отсутствие полного и приемлемого формального определения понятия визуального образа в произвольной визуальной среде. В работе представлена идея подхода к такому определению последовательными приближениями, и описаны три приближения.

Ключевые слова: визуальный образ, распознавание изображений, аффинные преобразования.

1. Введение

Подавляющая часть решений (успешных) в рамках компьютерного зрения сделана на основе эвристики — здравого смысла и изобретательности применительно к частным особенностям конкретной значимой для практики задачи. Приемлемой общей, глубокой и математизированной теории при этом не возникает, и во многом потому, что нет принятого в полной мере определения зрительного образа. Интуитивное понимание того, что есть зрительный образ, существует, а формального, математического определения нет. На том же содержательном, интуитивном уровне ясно, что зрительный образ — понятие сложное, многоплановое и многоуровневое. Это не одиночная, конкретная картинка, хотя и в этой картинке присутствуют черты образа, которому принадлежит картинка, и, в общем случае, может быть, и не единственного. Образы могут пересекаться, включать один другой, содержать в себе то, что можно назвать подобразами, а это выстраивает их в сложную систему взаимозависимостей. К тому же возникают образы из взаимодействия с окружающей средой, и алгоритмы такого возникновения — во многом загадка. Переход к доказательному, теоремному уровню исследования образов упирается, тем самым, в отсутствие приемлемого формального определения образа. Вряд ли можно решить задачу построения такого определения некоторым еди-

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

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

новременным усилием. Здесь предприняты некоторые начальные шаги в этой задаче.

2. Об определении изображения и об идее подхода к определению образа

Изображением называем конечное (непустое) множество точек на плоскости. Обосновываем это тем, что любую фигуру можно «аппроксимировать» конечным множеством точек (рис. 1), которые уже сами по себе делают фигуру вполне узнаваемой.

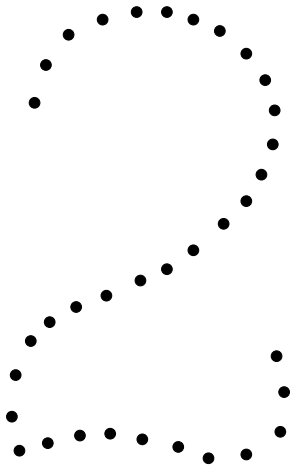

Рис. 1.

Если точек много, то такая совокупность точек практически неотличима от исходной фигуры. Так же можно представлять и полутоновые, черно-бело-серые изображения, при этом разная плотность точек в разных частях изображения дает разные оттенки «серого цвета». Как известно, цветное изображение можно представлять как наложение трех монохроматических (аналогов черно-бело-серых) изображений. Это означает, что совокупностями точек можно представлять и цветные изображения. Трехмерные изображения — точки в трехмерном евклидовом пространстве. Наконец, трехмерный мир в динамике можно рассматривать как четырехмерное изображение (последовательность трехмерных сцен). Далее рассматриваются двумерные изображения, но сказанное несложно обобщается и на случаи большей размерности.

Частью изображения A называем любое непустое подмножество B его точек. Любое изображение B', аффинно эквивалентное изображению B, называем подизображением изображения A.

Среда S у нас — произвольное изображение, состоящее из $N(N \geq 1)$ точек. Число частей среды — $\left(2^N - 1\right)$. Множество частей (т.е. изображений) обозначим через S'. Образ из среды S будем трактовать как некоторую группу изображений из S', т.е. задавать его перечислением всех возможных примеров изображений образа в данной среде S. Обозначим множество всех возможных групп через $S^\star$. Разумеется, далеко не все из этих групп — образы в содержательном понимании. Тем самым, задача состоит в том, чтобы «ужать» множество $S^\star$ до совокупности групп таких, которые уже можно с приемлемой степенью убедительности трактовать как образы. В этом и состоит подход к определению понятия образа в данной работе. Приближений к понятию образа предполагается несколько, в этой статье описано первое, второе и третье (отдельный образ, полный образ и замкнутый образ).

Ясно, что при таком подходе визуальный образ (и система образов) зависят от среды. Содержательная интерпретация этому была дана еще в давней работе [3]. В дальнейшем могут возникнуть интересные вопросы, связанные как с вариациями в системах образов, возникающих при средах разного типа, так и с константными особенностями такого рода систем.

3. Отдельная группа (отдельный образ)

Введем сквозной для последующего изложения пример: представим для наглядности среду S как «хаос» на плоскости разных фигур — цифр, букв и пр. — по разному расположенных, разных по размерам, ориентации, пересекающихся, имеющих общие части и пр. Тогда среди групп множества $S^\star$ будут как «осмысленные», т. е. группы, например, «двоек», или «троек», так и «бессмысленные» — сочетания одновременно и «двоек», и «троек», их частей, и многого другого. Вот эти бессмысленные сочетания и надо отсеять, основываясь на некоторых далее вводимых принципах.

Перенумеруем точки среды S с единственным условием: разные точки — разные номера. Полагаем, что других точек изображений, кроме точек среды S, на плоскости нет, т.е., в частности, копий изображений из S' не создается. Каждое изображение из множества S' может быть подвергнуто аффинным преобразованиям, но при этом всегда на плоскости присутствуют только N точек (возможно, частью преобразованных) исходного изображения S.

Два изображения назовем непересекающимися или отдельными, если у них нет общих точек (т.е. пересечение множеств номеров точек этих двух изображений пусто).

Пусть G — произвольная группа из $S^\star$, и в ней есть пара изображений A и B с общей непустой частью x. Тогда при аффинном преобразовании изображения A либо аффинно преобразуется и его часть x (отдельно от

остальных точек изображения B, тем самым B как таковое исчезает), либо A и B преобразуются (исключая частные вырожденные случаи) как совокупное единое изображение. И то, и другое с содержательной точки зрения не приемлемо. Отсюда возникает требование попарной непересекаемости изображений в группе, такие группы назовем отдельными. Множество всех отдельных групп обозначим через S^+, это и есть первое сокращение множества $S^\star$ всех групп.

В последующем будем рассматривать аффинные преобразования изображений из среды S. При таких преобразованиях часть точек изображения A может совпасть с частью точек изображения B. Такие точки называем кратными, сохраняем приписанными кратной точке номера слившихся точек, и при аффинных преобразованиях изображений A и B считаем их преобразующимися независимо друг от друга. На исходном изображении S кратных точек нет, после некоторых преобразований изображений среды они могут появиться. Пусть при этом исходная S преобразована в некоторую S''. Считаем, что у нас есть операция, которая, при необходимости, восстанавливает по S'' обратными преобразованиями исходную среду S.

4. Содержательное (неформальное) построение понятия футляра для группы изображений

Дано изображение X. Обозначим через $W(X)$ выпуклую оболочку для X. Известно [5], что для выпуклого множества $W(X)$ существует единственный наименьший по площади эллипс, его вмещающий. Центр этого эллипса назовем центром изображения X, а длину большей оси — размером (для вырожденного случая, когда все точки из X расположены на прямой, эллипс с очевидностью превращается в отрезок прямой, центр — середина отрезка).

Два изображения A и B называем независимыми, если $W(A)$ и $W(B)$ не имеют общих точек.

Множество всех частей изображения X обозначим через X'. Рассматриваем покрытие P_X изображения X попарно независимыми частями из X' (такое всегда есть). Части называем также кусками изображения X. Размером покрытия называем размер наибольшего куска в нем. Ясно, что возможных покрытий — конечное множество.

Уместно сделать следующее пояснение к дальнейшему. Построение модели — это всегда нечто такое, что первоначально возникает главным образом на основе интуиции, правдоподобных рассуждений. А уже затем на этой первоначальной основе строится совокупность определений (формальных, или, для начала, полуформальных), позволяющих прово-

дить доказательные рассуждения, т.е. получать утверждения (теоремы) о свойствах модели. Эвристика, в немалой степени присутствующая в распознающих системах — это тоже правдоподобные рассуждения. Но она, как правило, только ими и ограничивается.

Далее представлены два пункта содержательных соображений, на которых в значительной мере основывается модель.

Понятие остова для отдельного изображения и для группы изображений. Предположим, в качестве гипотезы, что для группы похожих изображений во всех них есть нечто общее, объединяющее эти изображения, и что мы назовем их остовом. Опишем возникновение этого понятия на содержательном уровне в нескольких шагах-приближениях, начиная с простого и довольно очевидного. Представим «двойку» в виде совокупности точек с кругами (рис. 2). Это основа, базовое изображение A. Трактуем его так: исходная

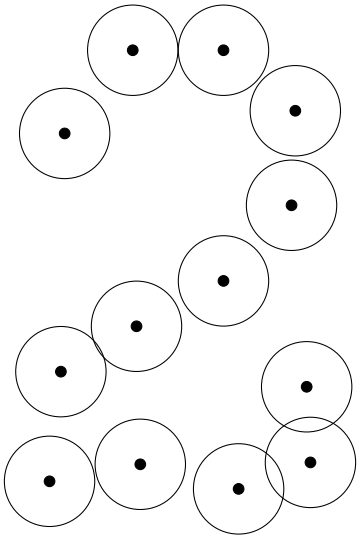

Рис. 2.

совокупность точек разбита на куски, причем такие, что тот единственный и наименьший по площади эллипс, который вмещает кусок, есть круг, причем все круги — одного радиуса, и их центры и образуют «двойку», которую мы видим на рисунке. Это несколько искусственный пример, потому что если реальную «двойку» разбить на куски, то минимальные по площади эллипсы этих кусков вовсе не обязательно будут именно кругами. Круги могут — немного — пересекаться. Вписанные же в круги куски (они на рисунке не представлены), как и следует из определений выше, пересекаться не могут. Предполагается, что при фиксированном

радиусе количество кругов наименьшее из возможных. Заменим теперь содержащееся в каждом круге множество точек (на рис. 2, напомним, они не показаны) на другое, но с тем же условием: выпуклая оболочка на этом множестве порождает кусок, вписанный в круг, причем центр куска совпадает с центром круга, куски из разных кругов, естественно, не пересекаются. Это второе изображение, ясно, отличается от первого. Аналогично можно породить целый класс изображений, обозначим их $A_1, ..., A_k$ (потенциально бесконечный), они разные (и разнообразные), поскольку в круги можно «вставлять» разные изображения, даже, например, те же «двойки», только маленькие по размерам. Изображение из центров кругов для каждого из изображений можно назвать остовом, он общий для всех изображений класса, и, полагаем для наглядности на данном этапе, присутствует явно в виде точек в каждом изображении, хотя в более общем случае точки остовов можно считать особыми, вспомогательными точками, не обязательно совпадающими с какими-то собственно точками изображений. Весь класс порожденных таким образом изображений можно рассматривать как определяющий преобразования для этих конкретных изображений более широкие, чем аффинные преобразования: каждая точка в пределах круга может быть преобразована в любую другую точку, но в пределах того же круга. Если в остове m точек, то его можно рассматривать как приближение m-го уровня для изображений класса (соответственно называть m-остовом). Остов есть «выжимка» из каждого такого изображения, и эта выжимка у них одинаковая (точнее: остовы аффинно эквивалентны, а в таком представлении, как на рис. 2, просто совпадают). Полагаем, что описанные изображения класса можно, с некоторой степенью убедительности, считать относящимися к одному образу. Сразу отметим, что если в исходном изображении n точек, то можно потенциально рассматривать классы изображений, порождаемых уровнями, начиная от 1 и до n.

Описанное было первым шагом. В следующем приближении можно расширить класс изображений m-го уровня, считая, что в любой из кругов базового изображения (рис. 2) может помещаться любое множество точек, не обязательно с центром именно в центре круга, но, конечно, внутри круга и без пересечений с другими кусками того же изображения. Если исходный рис. 2 рассматривать как футляр, то такие новые изображения можно трактовать как наполнение футляра, и точки наполнения, по сути, ограничены только тем условием, что они находятся внутри кругов. Полагаем, однако, что если есть набор кусков A с m-остовом a (базовое изображение, рис. 2), и набор кусков B с m-остовом b, то a и b находятся в оптимальном взаиморасположении, то есть точки изображений a и b максимально «придвинуты» друг к другу (в предыдущем, более частном случае, эти остовы просто совпадали). Этим исключается, например,

такой случай, когда точки из a и b не совпадают, но только потому, что сдвинуты по отношению друг к другу параллельным переносом. Итак, куски изображения B из наполнения у нас уже не обязательно равновелики по размерам (ранее равновеликость имела место по причине вписанности кусков в круги одинакового радиуса на рис. 2). Но для кусков исходного изображения A это условие пока сохраняется. Равновеликость важна, ибо в противном случае разбиение трудно считать представляющим форму исходного изображения. Отойти от условия равенства по размерам кусков (для A), сохраняя смысл равновеликости, можно так: пусть разбиение на куски с центрами $a_1, ..., a_m$ таково, что существует R такое, что каждый кусок находится внутри круга радиуса R (и с центрами в $a_1, ..., a_m$), и круги эти не пересекаются (это условие далее уточняется). Смысл изменения в том, что теперь неважно, какого размера собственно кусок исходного разбиения (он может быть даже точкой), но кусок находится внутри круга, одинакового по радиусу с другими кругами, и эти круги не пересекаются. Такой набор кругов с центрами в точках остова называем также футляром (порожденным данным изображением). Уточним условия на радиус кругов и возможное их пересечение. Полагаем, что ни один круг не включает центр другого круга в качестве внутренней точки (т.е. максимальный радиус равен минимальному расстоянию между точками остова). Содержательный смысл этого условия (ограничения) очевиден. (Пример: пусть точка a — центр первого круга, и этот круг включает точку b — центр второго круга.Тогда a можно преобразовать в точку b, и затем в любую точку второго круга, в том числе, и за пределами первого круга, что, по смыслу, недопустимо). Круги могут соприкасаться, и даже пересекаться, но к внутренним точкам круга (усеченного) относим только большую часть круга по соответствующую сторону от отрезка — границы пересечения. Точки наполнения футляра — только внутренние точки (возможно усеченных) кругов.

Как совместить привязку футляров к конкретным изображениям в среде с тем, что мы декларируем рассмотрение с точностью до аффинных преобразований. Мы намереваемся рассматривать изображения при распознавании с точностью до аффинных преобразований, т.е. безотносительно к параллельным переносам, преобразованиям симметрии, вращениям, изменениям в размерах, сжатиям, растяжениям и любым их комбинациям. Это в какой-то мере соответствует тому, что мы рассматриваем именно форму фигур, если говорить о фигурах, а не какие-то не имеющие отношения к форме обстоятельства, связанные с внешними системами координат [1, 2, 4]. Вместе с тем известно, что в целом аффинные преобразования не сохраняют форму фигур: чрезмерными сжатиями или растяжениями можно сделать фигуру неузнаваемой в сравнении с оригиналом. Тем самым, эта чрезмерность должна быть ограничена. Но

чем и как, в каких пределах, и чем эти пределы должны определяться? Содержательные рассмотрения выше были явно привязаны к «месту», т.е. к конкретному изображению, которое мы называли базовым. Конкретными были и размеры кусков, и радиусы кругов. Как это совместить с рассмотрением с точностью до аффинных преобразований? Положим в нашей среде S есть группа G изображений «двойки», безусловно вложимых аффинными преобразованиями в футляр A на рис. 2, но разные по размерам, положению на плоскости, с локальными изменениями, и пр., в том числе, для примера, может присутствовать и изображение X, аффинно эквивалентное с изображением из центров кругов на рис. 2, однако сжатое к некоторой прямой с таким коэффициентом, что все его точки выстроились практически в одну прямую. Ясно, что узнать в этом множестве точек X остов из рис. 2 невозможно, хотя они и аффинно эквивалентны. Так вот в наших модельных построениях должно быть учтено, с одной стороны, то, что этот X с остовом на рис. 2 аффинно «совпадают», с другой — то, что уж примером формы «двойки» X явно служить не может. Изображения группы G — это конкретные наборы точек в разных частях среды S. С содержательной точки зрения они в разной степени соответствуют тому, чтобы называться типичной двойкой, эталоном. Мы должны выделить изображение, наиболее достойное того, чтобы считаться эталоном в группе, причем, напомним, безотносительно к размерам, ориентации и пр., к сжатию-растяжению в определенных пределах, притом что пределы эти должны возникать «внутри» модели, а не задаваться извне. Мы будем пробовать на роль эталона группы поочередно все изображения группы. Возьмем, для начала, изображение A на рис. 2. Положим, к виду рис. 2 оно приводится разбиением исходного множества точек на m равновеликих кусков, где m — число кругов на изображении рис. 2, и центры кусков — точки изображения рис. 2. Изображение из центров кусков обозначим через a^+. Далее для произвольного изображения A^0 из группы G осуществляем все возможные его разбиения на m кусков (вообще говоря, уже не заботясь о равновеликости), строим (для каждого разбиения) точечное изображение a_0^+ из центров этих кусков, и укладываем аффинными преобразованиями изображение a_0^+ на изображение a^+. Если укладка (оптимальное взаиморасположение) такова, что при этом и каждый соответствующий кусок оказывается внутри соответствующего круга, то изображение A_0 называем приемлемым для эталона A (при уровне дробности m эталона). Поочередно проверяем на приемлемость все изображения группы, и поочередно при каждом изображении, рассматриваемым в качестве эталона. Если приемлемости нет в каждом из этих рассмотрений, то уровень m дробности называем не адекватным группе. Наибольший уровень адекватности группы называем ее дробностью, он и служит основной характеристикой «одно-

родности» группы, близости по форме составляющих ее изображений. Чем ближе по форме изображения, тем, полагаем, больше дробность. Предельный случай — когда все изображения аффинно эквивалентны — даст максимально возможную в этом случае дробность, равную числу точек в каждом изображении. Минимальная дробность, и она есть всегда, для любой группы, равна единице. Разнородность изображений в группе понижает дробность: ясно, то если, например, в группу «двоек» добавить «четверку», то дробность понизится. Нетрудно видеть, что в этих построениях хоть и используются в целом аффинные преобразования, но они все же ограничены, в известной степени, примерами, т.е. изображениями группы G. И это разумно, ибо возможных аффинно преобразованных изображений континуум (их можно назвать своеобразными «фантазиями»), но их приемлемость мы связываем с конечным «опорным» множеством, т.е. множеством конкретных примеров из данной среды S, с изображениями группы G (это есть представленная нам «реальность», в отличие от фантазий).

5. Искомое (оптимальное) взаиморасположение изображений

Пусть изображение A состоит из точек $a_1, ..., a_n$, изображение B — из точек $b_1, ..., b_n$, ψ — одно из возможных взаимно однозначных соответствий между точками изображений A и B, которым точке a_i из A сопоставляется точка $b_{\psi(i)}$ из B $(i = 1, ..., n)$. Обозначим через $B^\star$ множество всех изображений, получаемых из B аффинными преобразованиями. Полагаем, что на B' из $B^\star$ сохраняется нумерация, порожденная изображением B, т.е. через b'_i на B' обозначается точка, в которую переходит при соответствующем преобразовании точка b_i из B.

Зададимся некоторым положительным числом ϵ. Обозначим через $\{B\}^\epsilon$ множество всех таких изображений B' из $B^\star$, для которых длина каждого отрезка $(b_i b'_i)$ $(i = 1, ..., n)$ не больше ϵ. Преобразования, переводящие изображения из $\{B\}^\epsilon$ друг в друга, назовем ϵ-аффинными. Содержательно их можно трактовать как ограниченные, локальные аффинные преобразования для B.

Дадим определение искомого (или оптимального) взаиморасположения: через $l_A(B')$ обозначим длину наибольшего из отрезков $\left(a_i b'_{\psi(i)}\right)$ $(i = 1, ..., n)$. Рассмотрим B_0 — некоторое изображение из $B^\star$, и ψ_0 — одно из взаимно однозначных соответствий между точками изображений A и B. Пусть существует такое ϵ_1, что для всех B' из $\{B_0\}^{\epsilon_1}$ и при всех биекциях ψ минимум величин $l_A(B')$ достигается на изображении B_0 и при биекции ψ_0. Пусть существует такое ϵ_2, что для всякой пары изоб-

ражений (A', B'_0), получаемой ϵ_2-аффинными преобразованиями пары (A, B_0) как целого, выполняется аналогичное свойство: для всех B'' из $\{B'_0\}^{\epsilon_1}$ и при всех биекциях ψ минимум величин $l_{A'}(B'')$ достигается на изображении B'_0 и при биекции ψ_0. Тогда B_0 называем искомым для изображения A (и взаиморасположение A и B_0 искомым), биекцию ψ_0 — искомым соответствием между точками в A и B.

Что есть в содержательной интерпретации искомое (оптимальное) расположение изображения B на A, например, в случае, если A «эталонное» изображение «двойки», а B — тоже «двойка», но несколько искаженная по форме? Тогда B_0 — расположенная аффинными преобразованиями на A искаженная «двойка», причем расположенная так, чтобы, несмотря на исходные искажения, максимально повторять своей формой форму неискаженной A, при этом — безотносительно к размерам, ориентациям и сжатиям-растяжениям исходной фигуры B. Параметр $l_A(B')$ и служит мерой несовпадения форм фигур. Для двух двоек он, предполагается, будет существенно меньше, чем, скажем, для «двойки» и «четверки». В [1] представлены теоремы, на основе которых можно находить оптимальное расположение конечной процедурой.

6. Футляры для групп изображений

Выше было дано содержательное описание футляра для изображения A (или порождаемого изображением A). Уточним его некоторыми определениями и ссылками на ранее полученные теоремы. Зададимся некоторым m из промежутка от 1 до n (n — число точек в A). Рассмотрим покрытие A' изображения A независимыми кусками $A_1, \ldots, A_m$ (таких покрытий — конечное множество). Центры кусков обозначим через $a_1, \ldots, a_m$, в целом они составляют изображение a^+, его называем остовом (m-остовом). Пусть R — наименьшее расстояние между точками $a_1, \ldots, a_m$, полагаем каждую из этих точек центром круга радиуса R, и каждую из точек куска A_i ближе к a_i, чем к другим центрам. Последнее значит, что если круги пересекаются (по отрезку прямой), то кусок должен быть внутри усеченного круга. Если R меньше половины размера покрытия, то A' называем неправильным. Далее, когда имеются ввиду футляры, рассматриваем правильные покрытия для A. Футляр изображения есть пара $<$изображение A, остов $O>$. Этой пары достаточно, чтобы по ней на изображении A построить систему кругов радиуса R с центрами в точках из O, определить границы-отрезки пересечения кругов (если пересечения есть), куски изображения A в каждом из кругов. Пример (вырожденный): изображение A состоит из точек $a_1, \ldots, a_n$, рассматриваем разбиение на n кусков, т.е. каждый кусок — точка, R равен минимальному расстоянию между точками.

Итак, построен m-футляр изображения A. Если такой футляр не единственный, то каждый рассматривается независимо.

Пусть теперь B' есть одно из возможных разбиений изображения B на m непересекающихся кусков (не обязательно правильное), с центрами $b_1, ..., b_m$ (обозначение в целом: b^+). Если мы теперь аффинными преобразованиями трансформируем изображение B' вместе с точками $b_1, ..., b_m$, то новые положения точек $b_1, ..., b_m$ будут по прежнему центрами преобразованных кусков. Это следует из теоремы 1.

Теорема 1. *Пусть дано изображение X из точек $x_1, ..., x_n$ и точка y — его центр. Пусть изображение X' из точек $x'_1, ..., x'_n$ есть аффинно преобразованное изображение X, и точка y' — его центр. Тогда изображения из точек $x_1, ..., x_n, y$ и точек $x'_1, ..., x'_n, y'$ аффинно эквивалентны.*

Доказательство. Утверждение означает, что при аффинных преобразованиях изображения X из точек $x_1, ..., x_n, y$ в изображение X' из точек $x'_1, ..., x'_n, y'$ центр y изображения X переводится в центр y' изображения X'. Действительно, допустим, что центр преобразованного изображения X' есть точка y'', отличная от y'. Точка y есть центр эллипса E минимального по площади, в который вписано изображение X, т.е. если обозначить площадь эллипса через $P(E)$, площадь выпуклой оболочки изображения X через $P(X)$, то отношение $P(E)/P(X)$ минимальное из возможных. Такой эллипс согласно [5] существует и единственен. Аналогично точка y'' есть центр наименьшего по площади эллипса E'' для изображения X'. При аффинном преобразовании эллипс E преобразуется в эллипс E' с центром в y'. Поскольку y' и y'' предполагаются разными, то и эллипсы E' и E'' тоже разные. Но при аффинном преобразовании отношение $P(E)/P(X)$ сохранится равным $P(E')/P(X')$, а это значит, что эллипс E' тоже минимален по площади для X' и единственен — пришли к противоречию. Итак, E' и E'' должны совпадать, а, значит, совпадают и y'' с y'. Теорема доказана.

□

Определим понятие вместимости для B' в m-футляр изображения A. Расположим изображение b^+ на изображении O искомым образом. Это преобразует и изображение B' в целом. Если при этом каждый кусок изображения B' окажется внутри соответствующего круга (возможно, усеченного) футляра изображения A, то говорим, что данное m-разбиение изображения B вместимо в этот футляр. Изображение B называем m-вместимым в изображение A, если хотя бы одно из его m-разбиений вместимо в хотя бы один из m-футляров изображения A.

Пусть группа G из S^+ состоит из изображений $A_1, ..., A_k$, и пусть u — наименьшее число точек в изображениях группы. Зададимся некоторым

m из промежутка от 1 до u и пусть существует хотя бы одно из изображений группы такое, что все $A_1, ..., A_k$ m-вместимы в некоторый футляр этого изображения. Тогда группу называем m-совместимой. Наибольшее значение m, для которого группа m-совместима, обозначаем через M и называем дробностью группы. Те из изображений группы, M-футляры которых вмещают все изображения группы, называем ее эталонами. Теперь каждая группа из S^+ снабжена характеризующим ее параметром — дробностью, набором эталонов, и, будем полагать, M-футлярами на эталонах.

7. Полные группы (полные образы)

Пусть G_1 и G_2 — группы из S^+ с одинаковой дробностью, и пусть множество изображений G_2 есть подмножество (собственное) группы G_1. Тогда говорим, что группа G_1 полнее группы G_2. Группу называем полной, если нет группы полнее, чем она. Таким образом, здесь понятие полной группы основывается на задании группы перечислением входящих в нее изображений.

Пусть G — группа, M' есть M-футляр одного из эталонов группы. По построению, все изображения группы вместимы в M'. Однако, в M' могут быть вместимы и другие изображения среды. Назовем совокупность всех вместимых в M' изображений максимальным наполнением футляра M'. Нетрудно видеть, что максимальное наполнение футляра M' есть полная группа. Итак, ранее для полной группы было возможно ее задание, как и для всех групп, перечислением всех входящих в группу изображений. Не очень удобный способ, если изображений много. Однако теперь для полной группы возможно ее задание каким-либо футляром эталона группы, и указанием, что группу составляют все вместимые в этот футляр изображения.

Как можно содержательно трактовать описанное выше, т.е. наличие полной группы G и ее подгрупп, неполных, но максимальное наполнение которых совпадает с G? Задание примеров изображений группами (множество S^+) означает, что эти подгруппы могут быть разными: где-то примеров больше, где-то меньше, причем для одного и того же образа. В среде, в разных ее частях, может быть много разных примеров одной и той же фигуры (возможно, с вариациями в контурах). Скажем, некто познакомился с изображением «двойки» на сравнительно небольшом множестве примеров (т.е. это одна из возможных групп в S^+. Он «выработал» футляр для этой группы и это позволит ему при появлении неизвестного изображения определить, вместимо оно в этот футляр или нет, то есть распознать изображение. Другой субъект «выработает» схожий футляр (иными словами, схожее понятие «двойки») на другом множестве ее

примеров (другая группа), но в целом, возможно, изображения первой группы вместимы в футляр второй и наоборот, что и означает, что их можно объединить в одну группу. В целом это представляет в модели то содержательно очевидное обстоятельство, что одному и тому же образу можно «обучиться» на разных примерах и из разных частей среды. Множество всех полных групп обозначаем через S^{++} и трактуем как множество образов во втором приближении.

Предшествующее определение футляра основывалось на делении на куски изображений группы и рассмотрении центров этих кусков. Это определение использует ясные алгоритмы построения футляра, его и эти алгоритмы можно назвать первичными. Далее – несколько более общие определения ранее введенных понятий.

Пусть A — изображение состоящее из точек $a_1, \ldots, a_u$, изображение O (далее называем его остовом) состоит из точек $o_1, \ldots, o_n$, причем n не больше u, и дано некоторое число r не большее наименьшего расстояния между точками остова. Пусть для a_i (i=1, ... ,u) ближайшая к ней точка из O единственна, обозначим ее через o_j, и расстояние между точками o_j и a_i меньше r. Точку a_i называем прилежащей к точке o_j. Совокупность точек из A, прилежащих к точке o_j, называем куском изображения A, прилежащим к точке o_j, обозначаем через A_j (j=1, ... ,n). Если теперь каждая из точек из A попадает в некоторый кусок, и все куски изображения A непустые, то A и O называем приемлемо взаиморасположенными. Совокупность кусков A_j ($j = 1, \ldots, n$) называем разложением изображения A по остову O. Пару <A,O> называем футляром (n-футляром), порожденным изображениями A и O. Пусть существует изображение B', являющееся аффинно преобразованным изображением B, и приемлемо взаиморасположеное с O. Тогда B называем вложимым в футляр <A,O>, а изображение B' – его вложением. Куски B'_1, ... ,B'_n называем разложением вложения B' на куски по футляру.

Нетрудно видеть, что ранее введенное понятие футляра, остова и вложимости изображения в футляр есть частный случай последних определений. В отличие от первичного футляра <A,O>, для которого остов O строится по изображению A, в обобщении остов O в паре <A,O> рассматривается как данный, данным является и взаиморасположение изображений A и O.

Пусть задана полная группа G, состоящая из изображений $g_1, g_2, \ldots, g_k$, задан футляр $< g_1, O >$, в который вмещены все соответственно преобразованные изображения группы G, которые обозначим через $g_1, g'_2, \ldots, g'_k$. Пусть имеется изображение X, про которое известно, что оно вмещено в футляр $< g_1, O >$. Тогда X совпадает с одним из изображений $g_1, g'_2, \ldots, g'_k$. Действительно, группа G полная, значит никаких изображений, отличных от $g_1, g_2, \ldots, g_k$, в футляр $< g_1, O >$

вместить нельзя. Не может изображение X быть и копией какого-либо из изображений $g_1, g'_2, \ldots, g'_k$, несколько по иному расположенной в футляре, поскольку каждое из исходных изображений $g_1, g_2, \ldots, g_k$ преобразуется, но не копируется.

Пусть G и G' — полные группы, причем G' — подгруппа (собственное подмножество) группы G. Говорим, что G' максимальная подгруппа, если не существует среди множества всех полных подгрупп группы G такой группы G'', что G' есть ее подгруппа. Такую максимальную подгруппу G' называем непосредственным подобразом (по вложению) образа G. Набор всех непосредственных подобразов образа G называем спектром (по вложению) этого образа.

8. Продолжаемые и замкнутые группы изображений

Полный образ может содержать своеобразные «внутренние», «продолжаемые» образы. Дадим соответствующие определения. Пусть O — остов футляра, состоящий из точек $o_1, \ldots, o_n$. Пусть задана полная группа G изображений, и $B_1, \ldots, B_t$ есть вложения изображений группы G в футляр. Тем самым $B_1, \ldots, B_t$ есть аффинно преобразованные изображения из группы G и каждое из них приемлемо взаиморасположено с O. Отметим, что каждое изображение из G может быть потенциально не единственным образом приемлемо взаиморасположено с O, но реализуется только одно такое взаиморасположение, т.е. каждое изображение из G представлено среди $B_1, \ldots, B_t$ только одним изображением. Через b_{ij} $(i = 1, \ldots, t; j = 1, \ldots, n)$ обозначим разложение на куски вложения B_i по футляру.

Пусть $o_{j1}, \ldots, o_{jh}$ — непустое подмножество точек остова O, и ${b_{j1}}^i, \ldots, {b_{jh}}^i$ есть куски разложения изображения B_i, прилежащие к точкам соответственно $o_{j1}, \ldots, o_{jh}$. Обозначим изображение из точек кусков ${b_{j1}}^i, \ldots, {b_{jh}}^i$ через g_i $(i = 1, \ldots, t)$.

Еще раз отметим, что каждое из вложенных в футляр изображений $B_1, \ldots, B_t$ есть аффинно преобразованное изображение из исходной полной группы G, соответственно каждое из g_i $(i = 1, \ldots, t)$ есть аффинно преобразованная часть некоторого изображения из G. Обозначим совокупность всех этих частей через Q и говорим, что группа Q есть часть образа G, порожденная частью $o_{j1}, \ldots, o_{jh}$ остова O. Если группа Q полная, то образ Q называем существенной частью образа G, порождаемой частью $o_{j1}, \ldots, o_{jh}$ остова O.

Тривиальный пример части (и существенной части) образа можно представить на основе рис. 2. Если считать, что наполнение этого фу-

тляра есть все фигуры «двойки» из рассматриваемой среды, то удаление одной точки из футляра приведет к тому, что оставшиеся точки футляра породят образ, являющийся частью образа «двойки» (причем почти неотличимой от «двойки»).

Трактовка групп G и Q состоит в том, что если в среде обнаружено изображение X, принадлежащее образу G, то с необходимостью имеется в среде изображение x, являющееся частью X, и принадлежащее образу Q. И наоборот, если обнаружено x из Q, то существует и X из G, частью которого это x является. Это можно трактовать и как прогнозирование X по x, и, наоборот — x по X. Существенная часть Q образа G может быть одновременно и существенной частью другого образа P. В этом случае образ Q называем перекрестком образов G и P. Перекресток может быть общей частью не только двух, но и большего числа образов Образ Z, не являющийся существенной частью никакого другого образа, назовем замкнутым.

Пусть O_1 и O_2 есть непересекающиеся части остова O замкнутого образа G из соответственно точек $o_{i1}, \ldots, o_{im}$ и $o_{j1}, \ldots, o_{jk}$. Пусть O_3 есть объединение множеств O_1 и O_2, и при этом O_3 есть собственное подмножество множества точек остова O. Пусть V_1, V_2 и V_3 есть группы изображений, порождаемых соответственно частями O_1, O_2, и O_3 остова O и группы G. Задано, что V_1 есть существенная часть образа G. Вопрос: является ли V_3 существенной частью образа G? Вопрос можно трактовать так: O_1 порождает существенную часть V_1 образа G, к O_1 добавляем точки множества O_2, возникает часть O_3 остова O. Будет ли порождаемая им группа V_3 изображений тоже существенной частью образа G? Это можно трактовать и как вопрос о расширении, продолжении существенной части V_1 образа G до существенной части V_3 того же образа.

Теорема 2. *Группа V_3 изображений является существенной частью образа G.*

Доказателъство. Надо показать, что группа V_3 изображений является полной. Предположим, что это не так, то есть помимо изображений из V_3 есть еще изображение X, приемлемо расположенное в отношении остова O_3, и не входящее в V_3. Оно будет состоять из двух непересекающихся частей : первая есть X_1, вмещенное в O_1, и вторая есть X_2, вмещенное в O_2. Поскольку — V_1 полная группа, то X_1 должно совпадать с одним из изображений этой группы. Рассмотрим изображение Z из G, частью которого является X_1. Оно содержит часть — обозначим ее через Y — вмещенную в O_3, и состоящую из двух частей — обозначим их через Y_1 и Y_2, вмещенные соответственно в O_1 и в O_2. При этом Y не совпадает с X, но Y_1 совпадает с X_1. Рассмотрим теперь изображение Z' с частью X вместо части Y. Изображения Z и Z' имеют общую часть (это часть

X_1, совпадающая с Y_1). Изображение Z' по построению принадлежит исходной группе G, чего не может быть, ибо группа G — полная. Это первое противоречие, к которому мы приходим. А второе состоит в том, что у двух изображений в получившейся группе G — у Z и Z' — имеется непустое пересечение, что противоречит отдельности рассматриваемых образов. □

Теорему 2 можно рассматривать как обоснование для названия существенной части V_1 образа G образом, продолжаемым в замкнутый образ G.

Из определения замкнутой группы следует, что каждая полная группа либо замкнута, либо нет. В последнем случае группа является продолжаемой (причем в общем случае, не в единственную) замкнутую группу. Совокупность всех групп, продолжаемых в данную замкнутую группу, называем ее шлейфом. Таким образом, все множество полных групп делится на два подмножества: замкнутых групп и групп из шлейфов замкнутых групп.

Пусть существенная часть Q замкнутого образа G порождена частью $o_{i1}, \ldots, o_{im}$ остова, и любое (собственное) подмножество множества $o_{i1}, \ldots, o_{im}$ уже не порождает существенную часть. Тогда часть Q называем минимальной или признаком образа G. Тем самым, применительно к образу G можно говорить о множестве признаков для него. Частный случай признака, когда m равно единице, называем меткой образа G. При поиске в среде какого-либо изображения X образа G задача может быть, очевидно, заменена на поиск признака, или даже метки образа G.

Множество всех замкнутых групп считаем очередным приближением к понятию визуального образа.

Список литературы

[1] Козлов В. Н., *Введение в математическую теорию зрительного восприятия*, М.: Издательство Центра прикладных исследований при механико-математическом факультете МГУ, 2007.

[2] Козлов В. Н., “Conclusiveness and Heuristics in Visual Recognition”, *Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications*, **24**:4 (2014), 1 - 7.

[3] Крушинский Л. В., Козлов В. Н., Кудрвцев В. Б., *О некоторых результатах применения математики к моделированию в биологии*, в сборнике «Математические вопросы кибернетики», 1988.

[4] Кудрявцев В. Б., Гасанов Э. Э., Подколзин А. С., *Введение в теорию интеллектуальных систем*, М.: Издательство «МАКС Пресс», 2006.

[5] Загускин В. Л., “Об описанных и вписанных эллипсоидах экстремального объема”, *Успехи математических наук*, **13**:6 (1958), 89–93.

On three initial approximations to the formal definition of a visual image in an arbitrary visual environment

Kozlov V.N.

Visual image recognition is one of the central task for intelligent systems. The advancement of mathematical and theorem-based research methods in this field is significantly hampered by the lack of a complete and acceptable formal definition of the concept of a visual image in any visual environment. This paper presents an approach to such a definition using successive approximations and describes three approximations.

Keywords: visual image, image recognition, affine transformations.

References

[1] Kozlov V. N., *Introduction to the Mathematical Theory of Visual Perception*, Publishing House of the Center for Applied Research at the Faculty of Mechanics and Mathematics of Moscow State University, Moscow, 2007 (in Russian).

[2] Kozlov V. N., “Conclusiveness and Heuristics in Visual Recognition”, *Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications*, **24**:4 (2014), 1–7 (in Russian).

[3] Krushinsky L. V., Kozlov V. N., Kudryavtsev V. B, *On Some Results of Applying Mathematics to Modeling in Biology*, Mathematical Issues of Cybernetics, 1988.

[4] Kudryavtsev V. B., Gasanov E. E., Podkolzin A. S., *Introduction to the Theory of Intelligent Systems*, Publishing House “MAKS” Press, Moscow, 2006.

[5] Zaguskin V. L., “On circumscribed and inscribed ellipsoids of extremal volume”, *Uspekhi Matematicheskikh Nauk*, **13**:6 (1958), 89–93 (in Russian).

Часть 3
Математические модели

Применение отрицания к сильно связным автоматам

Д. О. Масленников[1]

Вводится понятие результата применения к инициальному автомату функции f, заданной на его выходном алфавите, как минимизированный инициальный автомат, реализующий определённую ограниченно-детерминированную функцию. Найдено достаточное условие его сильной связности.

Также введено понятия остова — неинициального автомата без выходной функции — и результата применения функции к нему, как неинициальный аналог предыдущего определения. Рассмотрены результаты применения отрицания к остовам определённого вида.

Для результата применения отрицания к сильно связному автомату с входным и выходным алфавитами $\{0, 1\}$ получены верхняя и нижняя оценки числа состояний, для чего было рассмотрено обобщение понятия пространства циклов на ориентированные графы.
Ключевые слова: конечный инициальный автомат, самомодифицирующийся конечный автомат, диаграмма Мура, граф, пространство циклов.

1. Введение

В этой работе рассмотрим модификацию инициального детерминированного конечного автомата: взяв за основание определение из [1] изменим алгоритм его так, чтобы на каждом такте выходной символ на диаграмме заменяется на другой в соответствии с некоторой функцией.

Различные модели автомата, которые при переходе могут менять функцию выхода, рассматривались и ранее. Наиболее схожая предложена Костером и Тейчем в [2]. Однако, в ней допускается также переписывание и функции переходов, а сами изменения определяются извне.

2. Определения и основные результаты

2.1. Применение функции к автомату

Определим точнее модификацию автомата, предложенную во введении. Пусть автомат $V = (A, Q, B, \phi, \psi, q_0)$, $f : B \to B$. Начиная работу с

[1] *Масленников Денис Олегович* — студент каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: denismaslenikov01@mail.ru.

Maslenikov Denis Olegovich — student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

состояния q_0, каждый такт проводятся следующие действия: как и в обычном автомате на вход поступает символ $a \in A$, совершается переход в новое состояние в соответствии с функцией ϕ и выводится символ $b \in B$ в соответствии с функцией ψ, а также изменяется диаграмма — символ b на совершённом переходе меняется на символ $f(b)$. Можем рассматривать данную конструкцию, которую назовём *результатом применения функции f к автомату V*, как словарную функцию — слову над алфавитом A из n символов сопоставляем слово над алфавитом B той же длины, которое будет выведено через n тактов.

Формализуем данное определение.

Обозначим $\Psi_{A,Q,B} = \{\psi : Q \times A \to B\}$ — множество функций вывода.

Пусть автомат $V = (A, Q, B, \phi, \psi, q_0)$, $f : B \to B$, тогда *результатом применения функции f к автомату V* называется словарная функция ω такая, что

$$\omega_t(x_1 \dots x_t \dots) = \chi(t)(q(t), x_t);$$

$$\chi : \mathbb{N} \to \Psi_{A,Q,B};$$

$$\chi(1) = \psi;$$

$$\chi(t+1)(q,x) = \begin{cases} f(\chi(t)(q,x)), & \text{если } q = q(t), x = x_t, \\ \chi(t)(q,x), & \text{иначе;} \end{cases}$$

$$q(t+1) = \phi(q(t), x_t),$$

$$q(1) = q_0.$$

По аналогии с тем, как в работе [1, с. 60] определяли изоморфизм между неинициальными автоматами называем изоморфизмом между инициальными автоматами $(A, Q, B, \phi, \psi, q_0)$ и $(A, Q', B, \phi', \psi', q'_0)$ биекцию $\xi : Q \to Q'$ такую, что $\phi'(\xi(q), x) = \xi(\phi(q, x))$, $\psi'(\xi(q), x) = \psi(q, x)$ и $q'_0 = \xi(q_0)$.

Утверждение 1. *Если V и V' изоморфны и $f : B \to B$, то результат применения f к ним совпадает.*

Доказательство. Пусть ξ — изоморфизм между V и V', тогда по индукции $q'(t) = \xi(q(t))$:
$q'(1) = q'_0 = \xi(q_0) = \xi(q(1))$,
$q'(t+1) = \phi'(q'(t), x_t) = \phi'(\xi(q(t)), x_t) = \xi(\phi(q(t), x_t)) = \xi(q(t+1))$.
И так же по индукции $\chi'(t)(\xi(q), x) = \chi(t)(q, x)$:
$\chi'(1)(\xi(q), x) = \psi'(\xi(q), x) = \psi(q, x) = \chi(1)(q, x)$.

$$\chi'(t+1)(\xi(q),x) = \begin{cases} f(\chi'(t)(\xi(q),x)), & \text{если } \xi(q) = q'(t) = \xi(q(t)), x = x_t, \\ \chi'(t)(\xi(q),x), & \text{иначе} \end{cases}$$
$$= \begin{cases} f(\chi(t)(q,x)), & \text{если } q = q(t), x = x_t, \\ \chi(t)(q,x), & \text{иначе} \end{cases}$$
$$= \chi(t+1)(q,x).$$

Получаем $\omega'_t(x_1 \dots x_t \dots) = \chi'(t)(q'(t), x_t) = \chi'(t)(\xi(q(t)), x_t) = \chi(t)(q(t), x_t) = \omega_t(x_1 \dots x_t \dots)$ ∎

Теорема 1. *Результат применения функции f к автомату $V = (A, Q, B, \phi, \psi, q_0)$ является ограниченно-детерминированной функцией, имеющей не более чем $n|B|^{n|A|}$ остаточных функций, где $n = |Q|$.*

Доказательство. Достаточно заметить, что для данную функцию можно задать автоматом, состояния которого — пары из $Q \times \Psi_{A,Q,B}$.

Зададим его явно — $\tilde{V}_f = (A, Q \times \Psi_{A,Q,B}, B, \tilde{\phi}, \tilde{\psi}, (q_0, \psi))$, где

$$\tilde{\phi}((q,\psi'),x) = (\phi(q,x), \psi''(q,\psi',x));$$

$$\psi''(q,\psi',x)(q',x') = \begin{cases} f(\psi'(q',x')), & \text{если } q' = q, x' = x \\ \psi'(q',x'), & \text{иначе.} \end{cases}$$

$$\tilde{\psi}((q,\psi'),x) = \psi'(q,x).$$

Проверим, что этот автомат реализует требуемую функцию. Для начала заметим, что

$$\psi''(q(t),\chi(t),x_t)(q',x') = \begin{cases} f(\chi(t)(q',x')), & \text{если } q' = q(t), x' = x_t \\ \chi(t)(q',x'), & \text{иначе.} \end{cases}$$

$= \chi(t+1)(q',x')$.

По индукции $\tilde{\phi}((q_0,\psi), x_1 \dots x_t) = (q(t+1), \chi(t+1))$.

Действительно, $\tilde{\phi}((q_0,\psi),\Lambda) = (q_0,\psi) = (q(1),\chi(1))$ и

$\tilde{\phi}((q_0,\psi), x_1 \dots x_t) = \tilde{\phi}(\tilde{\phi}((q_0,\psi), x_1 \dots x_{t-1}), x_t) = \tilde{\phi}((q(t),\chi(t)), x_t) =$

$= (\phi(q(t), x_t), \psi''(q(t),\chi(t),x_t)) = (q(t+1),\chi(t+1))$.

Имеем $\tilde{\psi}((q_0,\psi), x_1 \dots x_t) = \tilde{\psi}(\tilde{\phi}((q_0,\psi), x_1 \dots x_{t-1}), x_t) =$

$\tilde{\psi}((q(t),\chi(t)), x_t) = \chi(t)(q(t), x_t) = \omega_t(x_1 \dots x_t \dots)$.

Таким образом, ω действительно реализуется автоматом $\tilde{V}_f$, имеющим $n|B|^{n|A|}$ состояний. ∎

В силу предыдущей теоремы будем далее рассматривать результат применения функции к автомату не как словарную функцию, а как реализующий её конечный автомат приведённого вида. Обозначим его V_f. При этом в доказательствах часто будет удобно рассматривать его, как минимизированный $\tilde{V}_f$.

Пример 1.
$V_{id} = V$.

Пример 2.
Пусть автомат V имеет диаграмму:

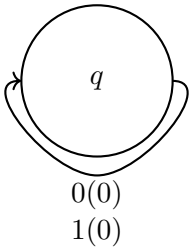

Рис. 1

Тогда состояние автомата $V_\neg$ определяется функцией выхода автомата V. Если $\psi(q, 0) = i$, $\psi(q, 1) = j$, то соответствующее состояние $V_\neg$ обозначим q_{ij}. Получаем диаграмму:

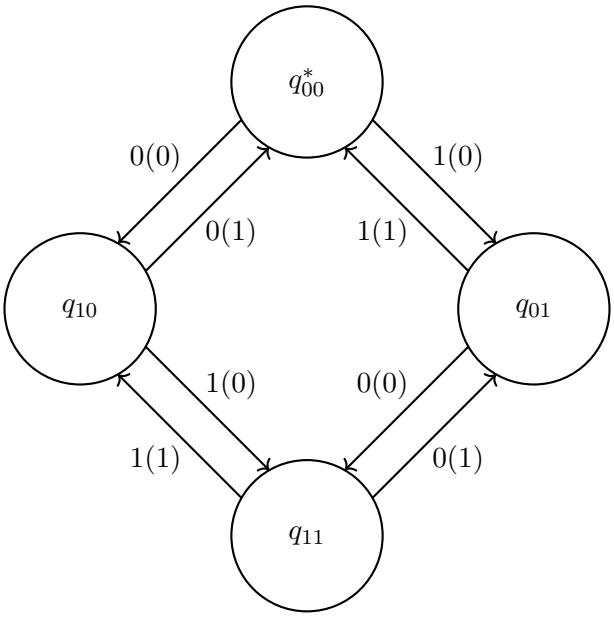

Рис. 2

Теорема 2. *Пусть V — сильно связный автомат, f — биекция. Тогда V_f — сильно связный.*

Доказательство. Автомат $V_f = (A, Q', B, \phi', \psi', q'_0)$ приведённый, следовательно для любого $q' \in Q'$ существует слово α_1 такое, что $\phi'(q'_0, \alpha_1) = q'$. Поскольку $V = (A, Q, B, \phi, \psi, q_0)$ сильно связный, существует слово

α_2 такое, что $q_0 = \phi(\phi(q_0, \alpha_1), \alpha_2) = \phi(q_0, \alpha_1\alpha_2)$. При получении на вход $(\alpha_1\alpha_2)^n$ по каждому переходу автомат V перейдёт кратное n число раз, так как $q(t)$ и x_t тогда имеют период (необязательно минимальный) $|\alpha_1\alpha_2|$. Поскольку f — биекция, найдётся такое n такое, что $f^n = id$. Тогда $\chi(1 + n|\alpha_1\alpha_2|) = \chi(1) = \psi$. Заметим также, что $q(1 + n|\alpha_1\alpha_2|) = \phi(q_0, (\alpha_1\alpha_2)^n) = q_0$. Применив это наблюдение к $\tilde{V}_f$, получаем, что слово $(\alpha_1\alpha_2)^n$ возвращает его в начальное состояние, а поскольку V_f — его минимизация, это верно и для него. Получаем $q'_0 = \phi'(q'_0, (\alpha_1\alpha_2)^n) = \phi'(\phi'(q'_0, \alpha_1), \alpha_2(\alpha_1\alpha_2)^{n-1}) = \phi'(q', \alpha_2(\alpha_1\alpha_2)^{n-1})$. Пусть $p' \in Q'$ — произвольное. Возьмём α_3 такое, что $\phi'(q'_0, \alpha_3) = p'$. Тогда $\phi'(q', \alpha_2(\alpha_1\alpha_2)^{n-1}\alpha_3) = \phi(q'_0, \alpha_3) = p'$, что означает сильную связность V_f. ∎

2.2. Применение функции к остову

Теперь введём неинициальный аналог результата применения функции к автомату. Поскольку функция вывода V влияет только на начальное состояние V_f, предполагаемый аналог не зависит от неё вовсе, поэтому вместо автомата будем использовать понятие, которое её не содержит.

Итак, назовём *остовом автомата* $V = (A, Q, B, \phi, \psi, q)$ или просто *остовом* четвёрку $\mathcal{V} = (A, Q, B, \phi)$. (Название никак не связано с остовным деревом графа.) Множество автоматов с остовом $\mathcal{V}$ обозначим $[\mathcal{V}]$. Диаграмму остова можно определить аналогично диаграмме автомата, но без выходных символов и начального состояния. Сильная связность для остова определяется так же, как и для автомата. (Автомат сильно связный тогда и только тогда, когда сильно связен его остов.)

Пусть $\mathcal{V} = (A, Q, B, \phi)$, $f : B \to B$. Тогда назовём *результатом применения функции f к остову $\mathcal{V}$* автомат приведённого вида $\mathcal{V}_f$ неотличимый от автомата $\tilde{\mathcal{V}}_f = (A, Q \times \Psi_{A,Q,B}, B, \tilde{\phi}, \tilde{\psi})$, где

$$\tilde{\phi}((q, \psi'), x) = (\phi(q, x), \psi''(q, \psi', x));$$

$$\psi''(q, \psi', x)(q', x') = \begin{cases} f(\psi'(q', x')), & \text{если } q' = q, x' = x \\ \psi'(q', x'), & \text{иначе.} \end{cases}$$

$$\tilde{\psi}((q, \psi'), x) = \psi'(q, x).$$

Пусть V — инициальный автомат. Тогда обозначим соответствующий ему неинициальный, как $\bar{V}$.

В [1, с. 22] сумма неинициальных автоматов $V = (A, Q, B, \phi, \psi)$ и $W = (A, Q', B, \phi', \psi')$, $Q \cap Q' = \varnothing$ определялась, как $V + W =$
$(A, Q \cup Q', B, \phi'', \psi'')$, где
$\phi''(q, a) = \phi(q, a)$, $\psi''(q, a) = \psi(q, a)$ при $q \in Q$ и $\phi''(q, a) = \phi'(q, a)$, $\psi''(q, a) = \psi'(q, a)$ при $q \in Q'$.

Определим сумму инициальных автоматов V и W, как неинициальный:
$V + W = \bar{V} + \bar{W}$.

Сумму нескольких автоматов обозначаем стандартно символом $\sum$.

Утверждение 2. *Пусть $\mathcal{V} = (A, Q, B, \phi)$ — остов, $f : B \to B$. Тогда $\mathcal{V}_f$ неотличим от $\sum\limits_{V \in [\mathcal{V}]} V_f$.*

Здесь нужно заметить, что V_f единственный с точностью до изоморфизма, поэтому в сумме можем взять автоматы такими, чтобы алфавиты состояний не пересекались и сумма была определена корректно.

Доказательство. Достаточно показать неотличимость $\tilde{\mathcal{V}}_f$ от $\sum\limits_{V \in [\mathcal{V}]} V_f$.

Состояние (q, ψ) автомата $\tilde{\mathcal{V}}_f$ неотличимо от начального состояния $\tilde{V}_f$, а значит и начального состояния V_f, где
$V = (A, Q, B, \phi, \psi, q) \in [\mathcal{V}]$.

Пусть q' — состояние автомата V_f. Из приведённости следует существование слова α переводящего автомат из начального состояния в q'. Как было сказано выше, начальное состояние неотличимо от состояния (q, ψ) автомата $\tilde{\mathcal{V}}_f$. Значит, $\tilde{\phi}((q, \psi), \alpha)$ неотличимо от q'. ∎

Теорема 3. *Пусть $\mathcal{V}$ сильно связный и f — биекция. Тогда $\mathcal{V}_f$ равен (с точностью до изоморфизма) сумме отличимых автоматов из множества $\{\bar{V}_f | V \in [\mathcal{V}]\}$.*

Доказательство. Заметим, что эта сумма отличается от суммы из утверждения 2 тем, что из неё исключены неотличимые от оставшихся автоматы, а с ними и неотличимые состояния. Следовательно, эти суммы неотличимы.

Осталось показать, что это автомат приведённого вида, то есть, что были удалены все неотличимые состояния. Предположим, что имеется пара неотличимых состояний. Поскольку все автоматы в сумме приведённого вида, то они могут лежать только в разных автоматах из суммы. При этом они сильно связные по теореме 2. Очевидно, что наличие неотличимых состояний в двух сильно связных автоматах влечёт неотличимость самих автоматов, что противоречит условию. ∎

Очевидно, что если автомат представляется, как сумма сильно связных, то только единственным образом. Эти слагаемые будем называть компонентами (сильной) связности.

Рассмотрим два примера, в которых функция отрицания $\neg$ применяется к сильно связному остову $\mathcal{V}$. Найдём количество и размер компонент связности в $\mathcal{V}_\neg$.

Утверждение 3. *Пусть остов $\mathcal{V}$ имеет диаграмму:*

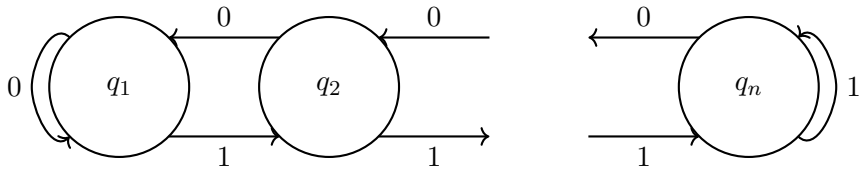

Рис. 3

Тогда $\mathcal{V}_\neg$ состоит из 2^{n-1} компонент связности по $n2^{n+1}$ состояния в каждой.

Доказательство. Убедимся, что $\tilde{\mathcal{V}}_\neg$ в данном случае приведённый. Возьмём состояния (q_1, ψ_1) и (q_2, ψ_2). Если $\psi_1 \neq \psi_2$, то $\psi_1(q_i, 0) \neq \psi_2(q_i, 0)$, или $\psi_1(q_i, 1) \neq \psi_2(q_i, 1)$. В первом случае различающим словом будет $1^{n-1}0^{n+1-i}$, а во втором — $0^{n-1}1^i$. (Это несложно проверить, подав нужное слово соответствующим автоматам из $[\mathcal{V}]$ и проследив их работу по алгоритму.) Если $\psi_1 = \psi_2$, а $q_1 \neq q_2$, то $\psi''(q_1, \psi_1, 0)(q_1, 0) = \overline{\psi_1(q_1, 0)} \neq \psi_2(q_1, 0) = \psi''(q_2, \psi_2, 0)(q_1, 0)$. То есть, $\tilde{\phi}((q_1, \psi_1), 0)$ и $\tilde{\phi}((q_2, \psi_2), 0)$ отличимы по доказанному выше, а значит отличимы и (q_1, ψ_1) и (q_2, ψ_2). Имеем, что $\mathcal{V}_\neg = \tilde{\mathcal{V}}_\neg$.

В силу того, что $\mathcal{V}_\neg$ — сумма сильно связных автоматов, из возможности перейти из одного состояния в другое состояния следует возможность вернуться из второго в первое, поэтому при исследовании на достижимость мы можем пренебречь направлением переходов на диаграмме $\mathcal{V}$ и перейти к аналогичному неориентированному графу с выделенной вершиной:

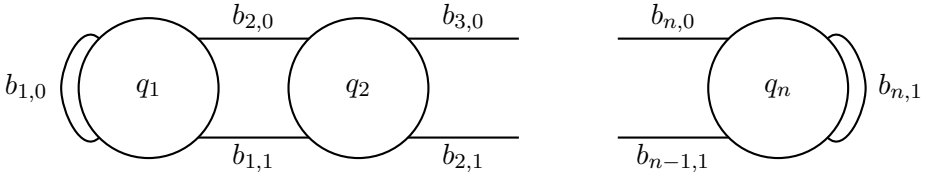

Рис. 4

Заметим, что можем всегда перейти к графу, который отличается только боковой петлёй:

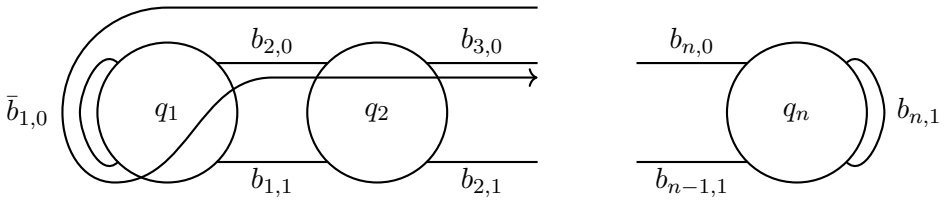

Рис. 5

Аналогично всегда можно изменить парные рёбра, не меняя другие:

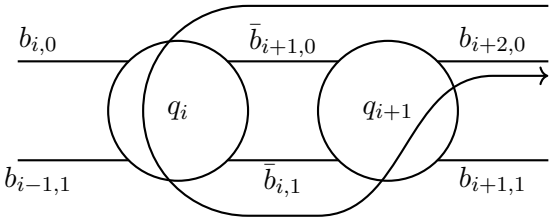

Рис. 6

Заметим, что при переходе из вершины в вершину по такому же закону меняется и сумма выходных символов парных рёбер по модулю 2. Пользуясь этими наблюдениями, можно перейти к графу без петель и кратных рёбер:

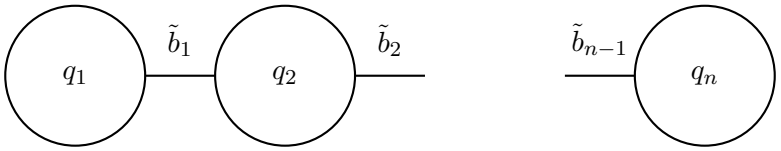

Рис. 7

Здесь $\tilde{b}_i = b_{i,1} + b_{i+1,0}(mod\ 2)$. Таким образом, полные графы с Рис. 4 разбиваются на классы по соответствию упрощённому графу с Рис. 7. При этом действие перехода из вершины в вершину на полном графе соответствует аналогичному на упрощённом, и все графы из одного класса взаимодостижимы. То есть, два полных графа взаимодостижимы тогда и только тогда, когда взаимодостижимы соответствующие упрощённые графы. Легко увидеть, что в каждом классе 2^{n+1} граф.

Заметим, что упрощённый граф можно перевести в граф с произвольным выделенным состоянием и при том единственный. В каждой компоненте достижимости можно выбрать каноничный граф с выделенным состоянием q_1. Таким образом, имеем в каждой компоненте n графов, и всего 2^{n-1} компонент, по количеству графов с выделенным состоянием q_1.

Имеем 2^{n-1} компоненты связности автомата $\mathcal{V}_\neg$ по $n2^{n+1}$ состояния в каждом. ∎

Пользуясь теоремой 3, мы можем переформулировать утверждение 3:

Следствие 1. *Если $\mathcal{V}$, как в утверждении 3, то в множестве $\{\bar{V}_\neg | V \in [\mathcal{V}]\}$ 2^{n-1} отличимых автомата и все автоматы в нём имеют $n2^{n+1}$ состояния.*

Далее используем обозначения: $m \mid n$ — m делит n, и $m \nmid n$ — m не делит n.

Утверждение 4. *Пусть остов $\mathcal{V}$ имеет диаграмму:*

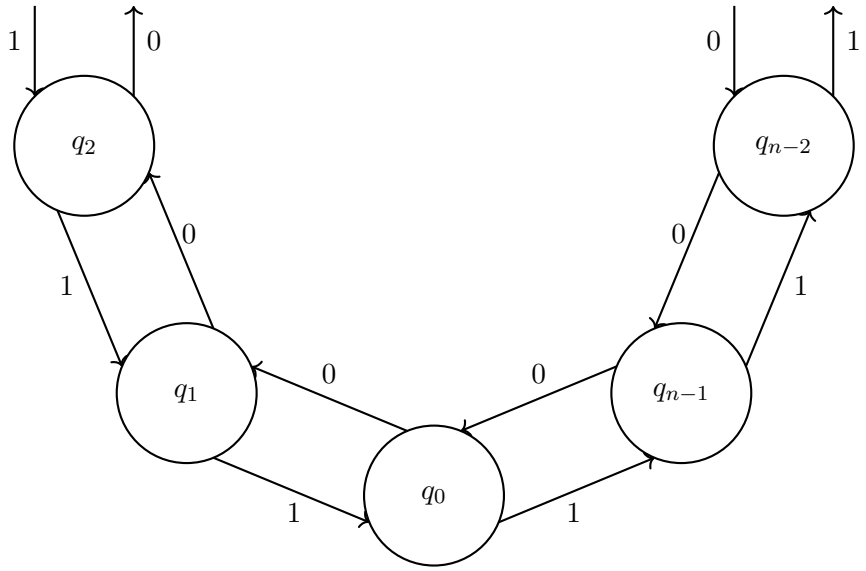

Рис. 8

$n = 2^k m$, $2 \nmid m$. Тогда компоненты связности $\mathcal{V}_\neg$ имеют $2^{n+k+1}d$ состояния, $d \mid m$, и есть ровно $\frac{1}{2^{k+1}d} \sum\limits_{s|d} \mu(s) 2^{2^k \frac{d}{s}}$ компоненты соответствующего размера, где μ — функция Мёбиуса.

Доказательство. Каждое состояния $\tilde{\mathcal{V}}_\neg$ неотличимо от начального состояния соответствующего автомата $V_\neg$, $V \in [\mathcal{V}]$. Однако, в $[\mathcal{V}]$ присутствуют изоморфные автоматы с изоморфизмами вида $\xi_j(q_i) = q_{i+j(mod\ n)}$, $j = \overline{0, n-1}$. По утверждению 1, результаты применения к ним отрицания неотличимы, поэтому из каждого класса изоморфизмов выберем каноничный автомат с начальным состоянием q_0. Очевидно, что результаты применения к ним отрицания отличимы. Таким образом, можем считать, что $\mathcal{V}_\neg$ имеет состояния вида (q_0, ψ).

Далее рассуждения аналогичны предыдущему примеру. В силу разбиения на компоненты сильной связности можем перейти к аналогичному неориентированному графу с выделенной вершиной.

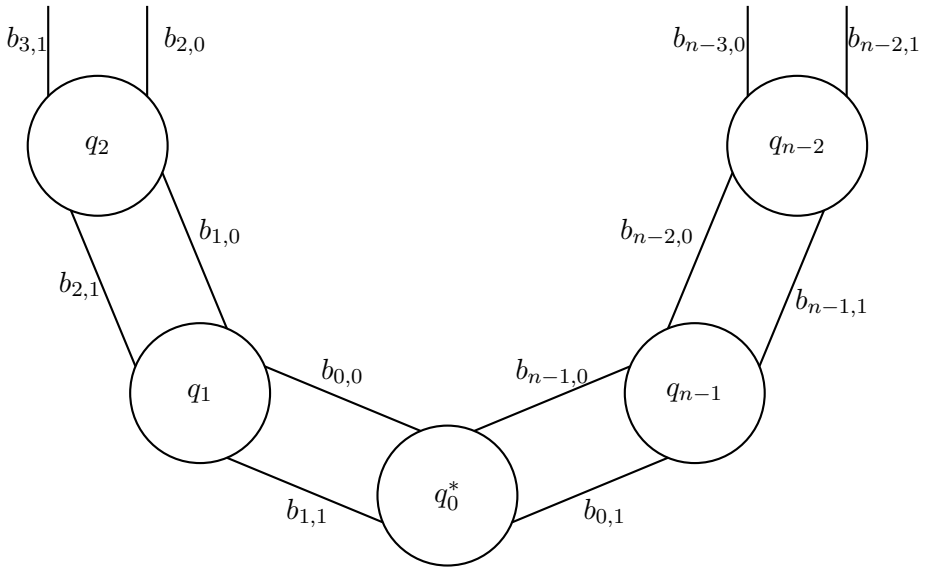

Рис. 9

Заметим, что всегда можем изменить парные рёбра, не меняя другие:

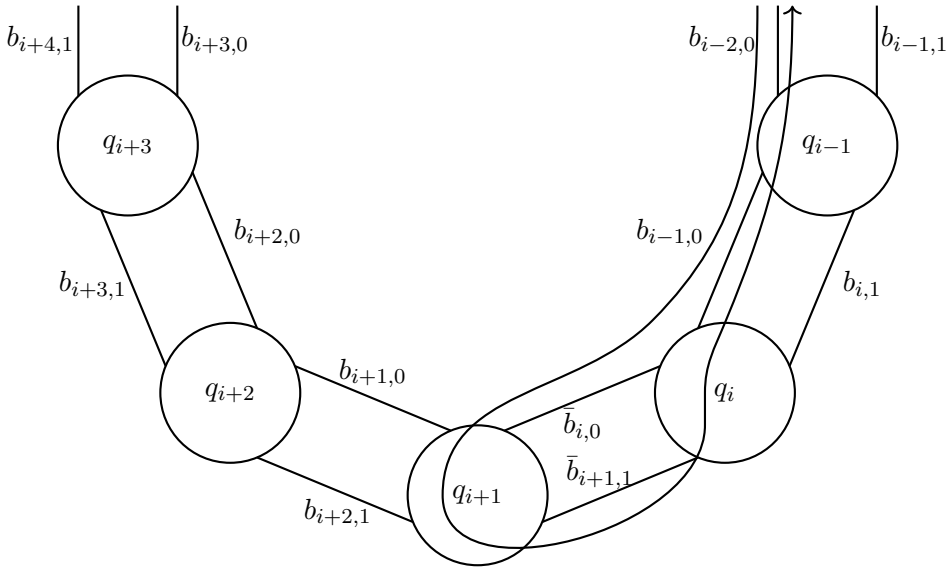

Рис. 10

Переходим к упрощённому графу без петлей и кратных рёбер:

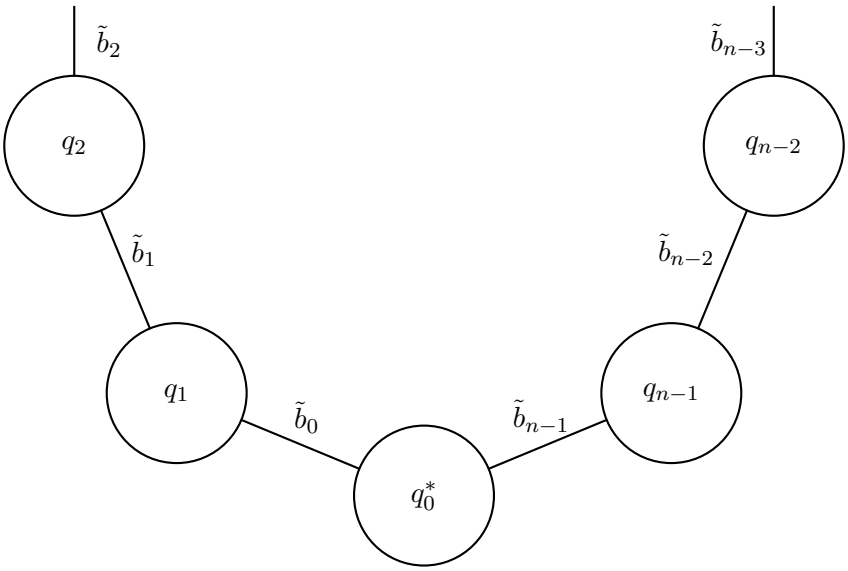

Рис. 11

Здесь $\tilde{b}_i = b_{i,0} + b_{i+1,1} (mod\ 2)$. Каждому упрощённому графу соответствует класс из 2^n полных графов, все графы в одном классе взаимодостижимы. В отличие от прошлого случая, в этом мы брали канонический автомат из класса изоморфизма, поэтому операция перехода сопровождается применением изоморфизма $\xi_{\pm 1}$. Заметим, что множество взаимодостижимых графов имеет циклическую структуру, периода $p \mid 2n$. При этом после n переходов в одну сторону получим «противоположный» граф. Следовательно, $p \nmid n$. Таким образом, $p = 2^{k+1}d$, $d \mid m$. Подсчитаем количество графов, которые лежат в множестве периода p (необязательно минимального). Если $p = 2n$, то это все $2^n = 2^{\frac{p}{2}}$. Если $p \neq 2n$, то заметим, что $\tilde{b}_i = \tilde{b}_{i+p} = \cdots = \tilde{b}_{i+n-\frac{p}{2}} = \bar{\tilde{b}}_{i+\frac{p}{2}} = \cdots = \bar{\tilde{b}}_{i+n-p}$, при этом $\tilde{b}_i$ не зависят друг от друга, $i = \overline{0, \frac{p}{2}}$. Имеем $2^{\frac{p}{2}}$ графов. Дальнейшие рассуждения повторяют задачу об ожерельях. ([3, с.9]) Обозначим через $N(p)$ число компонент связности периода (минимального) p. $\sum\limits_{s|d} 2^{k+1} s N(2^{k+1}s) = 2^{2^k d}$. Применим формулу обращения Мёбиуса. (Строго говоря, выражение под суммой и выражение справа должны быть определены на всём $\mathbb{N}$, а не только на делителях m, но можно заметить, что следующее равенство не зависит от того, как они были доопределены.) Получаем $2^{k+1} d N(2^{k+1}d) = \sum\limits_{s|d} \mu(s) 2^{2^k \frac{d}{s}}$. Итак, $\mathcal{V}_\neg$ имеет $\frac{1}{2^{k+1}d} \sum\limits_{s|d} \mu(s) 2^{2^k \frac{d}{s}}$ компонент связности по $2^{n+k+1}d$ состояния. ∎

Пользуясь теоремой 3, мы можем переформулировать утверждение 4:

Следствие 2. *Если $\mathcal{V}$, как в утверждении 4, то в множестве $\{\bar{V}_\neg | V \in [\mathcal{V}]\}$ автоматы имеют $2^{n+k+1}d$ состояния, $d \mid m$, и есть ровно $\frac{1}{2^{k+1}d}\sum\limits_{s|d}\mu(s)2^{2^k\frac{d}{s}}$ автомата с соответствующим числом состояний.*

2.3. Оценки на число состояний

Воспользуемся понятием пространства циклов и его размерностью. ([4, с. 203, 208-211], [5, с. 23-27]) Пусть G — граф. V_G, E_G — множества его вершин и рёбер соответственно, $c(G)$ — число его компонент связности. Пространство рёбер $W_E(G)$ — множество всех подмножеств E_G. На $W_E(G)$ задана операция:

$$E_1 \oplus E_2 = (E_1 \setminus E_2) \cup (E_2 \setminus E_1).$$

$W_E(G)$ — векторное пространство над $\mathbb{Z}_2$. Рассматриваем циклы, как множество рёбер, через которые они проходят. Пространство циклов $W_C(G) \subseteq W_E(G)$ — линейная оболочка множества циклов. Размерность $W_C(G)$ — циклический ранг $\beta(G) = |E_G| - |V_G| + c(G)$.

Распространим теорию на ориентированные графы. Пусть G — ориентированный граф. V_G, E_G, и $W_E(G)$ определяем аналогично, $W_C(G)$ определяем так, как если бы граф был неориентирован. $\bar{W}_C(G) \subseteq W_E(G)$ — линейная оболочка контуров (ориентированных циклов).

Теорема 4. *Пусть G — ориентированный сильно связный граф. Тогда $\bar{W}_C(G) = W_C(G)$.*

Доказательство. $\bar{W}_C(G) \subseteq W_C(G)$, так как порождающее множество $\bar{W}_C(G)$ входит в порождающее множество $W_C(G)$.

Далее в доказательстве мы допускаем самопересечение цикла, как сумму его рёбер. Например, цикл $\{a, b, c, d, e, c\}$ приведённого ниже графа будет равен циклу (неориентированному) $\{a, b, e, d\}$.

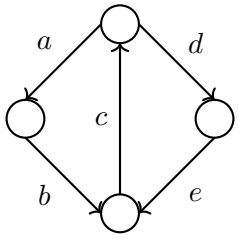

Рис. 12

Заметим, что контур с пересечениями можно разложить в сумму контуров без пересечений: пусть совпадают конечные вершины рёбер e_i и e_j, $i < j$, тогда

$$\{e_1, e_2, \dots e_1\} = \{e_1, e_2, \dots, e_i, e_{j+1}, \dots, e_1\} \oplus \{e_{i+1}, e_{i+1}, \dots, e_j\}.$$

Применяем это разложение, пока не избавимся от пересечений. Так в примере выше $\{a,b,c,d,e,c\} = \{a,b,c\} \oplus \{d,e,c\}$.

В цикле без самопересечений можно выделить участки постоянного направления. Между ними — вершины из которого выходит 2 или 0 рёбер этого цикла. Обозначаем v_i^- и v_i^+ соответственно. Если $v_1, v_2, \ldots, v_n, v_1$ — последовательность вершин цикла, то $v_1^-, v_1^+, v_2^-, v_2^+, \ldots, v_m^-, v_m^+, v_1^-$ — его подпоследовательность, и в таком случае есть $2m$ участка — по m на каждое направление, которое идёт от v_i^- к v_{i-1}^+ или v_i^+.

Докажем, что любой цикл без самопересечений лежит в $\bar{W}_C(G)$ индукцией по m. Если $m = 0$, то цикл — контур. Пусть для циклов с не более чем $2m$ участками доказано, что они лежат в $\bar{W}_C(G)$. Рассмотрим цикл E с $2(m+1)$ участком. Возьмём путь без самопересечений $e_1, e_2, \ldots, e_l$ от v_i^+ до некоторого v_j, не лежащим между v_i^+ и v_{i+1}^+ такой, чтобы он больше нигде не пересекал цикл, кроме может быть, между v_i^+ и v_{i+1}^+. (Такой есть в силу сильной связности.)Тогда E можно представить в виде суммы двух циклов — один из них является суммой этого пути и дугой цикла от v_j до v_i^+, другой — суммой пути и второй дуги от v_j до v_i^+. Рассмотри в отдельности один из этих циклов. (Второй полностью аналогично.) Если путь $e_1, e_2, \ldots, e_l$ не пересекает дугу от v_j до v_i^+, то это цикл без самопересечений. Легко видеть, что его число участков меньше, чем у исходного, так путь будет продолжением участка смежного с v_i^+, а значит не породит новый при том, что как минимум второй смежный с v_i^+ участок в этот цикл не войдёт. Если же путь пересекает дугу, то, как мы условились при его выборе, он может пересекать его только на участке смежном с v_i^+. Пусть v_k — последнее пересечение. Разложим цикл в сумму двух слагаемых. Первое — сумма пути от v_k до v_j и дуги от v_j до v_k, по рассуждениям аналогичным тем, что были выше, это будет цикл с меньшим числом участков. Второе — сумма пути от v_i^+ до v_k и дуги от v_i^+ до v_k. Это будет контур, возможно с пересечениями, а как показано выше, его можно разложить в сумму контуров без пересечений. Таким образом, E раскладывается в сумму циклов с не более чем $2m$ участками, а по предположению индукции они лежат в $\bar{W}_C(G)$. Следовательно, $E \in \bar{W}_C(G)$.

Таким образом, порождающее множество $W_C(G)$ лежит в $\bar{W}_C(G)$, а значит $W_C(G) \subseteq \bar{W}_C(G)$. ∎

Обозначим $Q(V)$ множество состояний автомата V.

Теорема 5. *Пусть $W(n)$ — множество сильно связных автоматов с n состояниями и $A = B = \{0,1\}$. Тогда верно, что*

$$\max_{V \in W(n)} |Q(V_\neg)| = n2^{n+1},$$

$$\min_{V \in W(n)} |Q(V_\neg)| \geqslant 2^{n+1}.$$

И при нечётных n достигается равенство.

Доказательство. Пусть V — сильно связный автомат с n состояниями. Докажем сначала, что $2^{n+1} \leqslant |Q(V_\neg)| \leqslant n2^{n+1}$. Можем рассматривать диаграмму V, как ориентированный граф, а функцию выходов соотнести с множеством рёбер диаграммы, к которым приписаны единицы. При этом переход по ребру суммирует его с этим множеством.

Заметим, что неориентированный цикл с пересечениями можно разложить в сумму циклов без пересечений аналогично тому, как это делается с контурами с разницей в том, что возможен случай, когда отсутствие пересечений по вершинам не гарантирует отсутствие пересечений по рёбрам — цикл вида $\{e, e\}$, но он равен пустому множеству, как сумма двух одинаковых рёбер.

Пусть q — состояние V. Введём обозначение: $q\Psi$ — множество достижимых состояний $\tilde{V}_\neg$ вида (q, ψ). Два произвольных слова, переводят $\tilde{V}_\neg$ из (q_0, ψ_0) в состояние из $q\Psi$ тогда и только тогда, когда они переводят V из q_0 в q. Сложим пути на диаграмме, соответствующие этим словам, и получим, что разность между двумя функциями — неориентированный цикл. В соответствии с наблюдением выше, его можно представить как сумму циклов без пересечений. То есть, все состояния из $q\Psi$ отличаются друг от друга по второй компоненте на элемент из $W_C(G)$.
$(\{\psi_1 \oplus \psi_2 | (q, \psi_1), (q, \psi_2) \in q\Psi\} \subseteq W_C(G))$

Теперь покажем, что мы можем прибавить произвольный контур к фиксированному ψ, $(q, \psi) \in q\Psi$. Выберем на контуре состояние q'. Пусть α — слово, которое нужно подать, что пройти по контуру. В силу сильной связности существуют β_1 и β_2 такие, что $\phi(q, \beta_1) = q'$ и $\phi(q', \beta_2) = q$. Тогда слово $\beta_1\alpha\beta_2\beta_1\beta_2$ суммирует ψ с данным контуром и возвращает автомат в состояние q. $(\bar{W}_C(G) \subseteq \{\psi \oplus \psi_1 | (q, \psi_1) \in q\Psi\})$

Применяя теорему 4 получаем

$$W_C(G) = \bar{W}_C(G) \subseteq \{\psi \oplus \psi_1 | (q, \psi_1) \in q\Psi\} \subseteq$$
$$\subseteq \{\psi_1 \oplus \psi_2 | (q, \psi_1), (q, \psi_2) \in q\Psi\} = W_C(G).$$

Имеем $W_C(G) = \{\psi \oplus \psi_1 | (q, \psi_1) \in q\Psi\}$.

Следовательно, $q\Psi = \{(q, \psi \oplus \psi_1) | \psi_1 \in W_C(G)\}$.

Итак, $|q\Psi| = |W_C(G)| = 2^{2n-n+1} = 2^{n+1}$. Очевидно, что все состояния из $|q\Psi|$ отличимы. Это даёт нижнюю оценку. Для верхней оценки достаточно домножить это число на число состояний.

Осталось показать, что для любого n существует случай, когда $|Q(V_\neg)| = n2^{n+1}$, и любого нечётного n случай, когда $|Q(V_\neg)| = 2^{n+1}$. Для этого достаточно обратиться соответственно к следствию 1 и к следствию 2, взяв $k = 0$ и $d = 1$. ∎

3. Заключение и выводы

Было приведено достаточное условие сильной связности результата применения функции к автомату — биективность функции и сильная связность автомата. Были получены оценки на число состояний результата применения в случае выполнения этого условия — оно экспоненциально зависит от числа состояний исходного автомата. Была описана структура двух частных случаев результата применения функции к остову относительно количества и размера компонент связности.

4. Благодарность

Выражаю благодарность своему научному руководителю А. П. Соколову.

Список литературы

[1] Кудрявцев В.Б., Алешин С.В., Подколзин А.С., *Введение в теорию автоматов*, Издательство Московского университета, 2019, 436 pp.

[2] M. Koster, J. Teich, "(Self-)reconfigurable Finite State Machines: Theory and Implementation", *Proceedings 2002 Design, Automation and Test in Europe Conference and Exhibition*, 2002, 559–566

[3] M. Lothaire, *Combinatorics on Words*, Second Edition, Cambridge University Press, 1997, 238 pp.

[4] Gross J. L., Yellen J., Anderson M, *Graph Theory and Its Applications*, Third Edition, Chapman and Hall/CRC, New York, 2018, 591 pp.

[5] Diestel R., *Graph Theory*, New York, 2005, 422 pp.

Applying negation to strongly connected automata
Maslenikov D.O.

The concept of the result of applying a function f, defined on its output alphabet, to an initial automaton is introduced as a minimized initial automaton implementing a certain boundedly deterministic function. A sufficient condition for its strong connectivity is found.

The concepts of a skeleton — a noninitial automaton without an output function — and the result of applying a function to it, as a noninitial analog of the previous definition, are also introduced. The results of applying negation to skeletons of a certain type are considered.

For the result of applying negation to a strongly connected automaton with input and output alphabets $\{0, 1\}$, upper and lower bounds for the number of states are obtained, for which a generalization of the concept of a cycle space to oriented graphs was considered.
Keywords: finite automaton, self-modifying finite state machine, Moore diagram, graph, cycle space.

References

[1] Kudryavtsev V.B., Alyoshin S.V., Podkolzin A.S., *Introduction to automata theory*, Moscow University Press, 2019 (in Russian), 436 pp.

[2] M. Koster, J. Teich, "(Self-)reconfigurable Finite State Machines: Theory and Implementation", *Proceedings 2002 Design, Automation and Test in Europe Conference and Exhibition*, 2002, 559–566

[3] M. Lothaire, *Combinatorics on Words*, Second Edition, Cambridge University Press, 1997, 238 pp.

[4] Gross J. L., Yellen J., Anderson M, *Graph Theory and Its Applications*, Third Edition, Chapman and Hall/CRC, New York, 2018, 591 pp.

[5] Diestel R., *Graph Theory*, New York, 2005, 422 pp.

Задача определения порядка для автоматов, чьи функции переходов и выходов принадлежат замкнутому классу Поста

Н. В. Муравьев[1]

Рассматривается задача определения порядка автомата Мили относительно операции суперпозиции.

Доказано разбиение решетки Поста замкнутых классов относительно разрешимости задачи вычисления порядка для соответствующих R-автоматов.

Ключевые слова: автоматы Мили, классы Поста, алгоритмическая разрешимость.

1. Введение

Определение порядка элемента в полугруппе является классической задачей в алгебре. Несмотря на то, что в общем случае она алгоритмически неразрешима даже в группе конечных автоматных функций [1], автору ранее удалось найти богатые классы линейных автоматов, для которых не только существует алгоритм вычисления порядка относительно суперпозиции, но и есть точная верхняя оценка порядка автомата, зависящая от его размерности и базового поля [2, 3, 4, 5].

Таким образом, задачу можно решить, если на входном-выходном алфавите и множестве состояний удается ввести структуру линейного пространства. Естественно задаться вопросом: «какие еще структуры могут позволить алгоритмически определять порядок автоматной функции?». В настоящей работе мы вводим на алфавитах и множествах состояний структуру булевого куба. А функции переходов и выходов будут «уважать» эту структуру, если они принадлежат какому-нибудь фиксированному замкнутому классу Поста. Показано, что относительно алгоритмической разрешимости задачи для соответствующих классов автоматов решетку Поста можно разбить на две группы: классы, порождающие автоматы с неразрешимой задачей определения порядка, и классы, порождающие автоматы с разрешимой задачей определения порядка.

[1] *Муравьев Никита Валерьевич* — аспирант каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: ne-ki-tos@yandex.ru.

Muravev Nikita Valerievich — graduate student, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

Полученные результаты уточняют границы разрешимости задачи определения порядка и дают нам новые нетривиальные классы автоматов, для которых она разрешима.

2. Технические леммы и основной результат

Рассматриваем инициальные автоматы Мили вида $G = (\Sigma, Q, \Sigma, \phi, \psi, q_0)$, чей входной-выходной алфавит является многомерным булевым кубом $\Sigma = E_2^n$, а множество состояний является произвольным подмножеством булевого куба $Q \subset E_2^m$.

Для замкнутого класса Поста $R \subset \mathbb{P}_2$ мы говорим, что автомат G является R-автоматом, если его функции переходов и выходов принадлежат этому классу $\phi, \psi \in R$. Нетрудно убедиться, что суперпозиция двух R-автоматов вновь будет R-автоматом.

Имеет место следующая

Теорема 1. *Задача определения порядка алгоритмически неразрешима в классе R-автоматов, если R содержит в себе один из следующих классов как подмножество: $F_2^\infty, F_6^\infty, D_2$. В противном случае задача определения порядка алгоритмически разрешима (рис. 1).*

Несмотря на то, что замкнутых классов бесконечное число, для доказательства этого результата нам понадобится рассмотреть лишь конечное число классов. Это следует из технических лемм, аналоги которых уже неоднократно встречались в литературе [6, 7]:

Лемма 1. *Рассмотрим замкнутые классы $R_1 \subset R_2 \subset \mathbb{P}_2$. Если не существует алгоритма определения порядка для R_1-автоматов, то его не существует и для R_2-автоматов.*

Доказательство. Лемма тривиально следует из того факта, что любой R_1-автомат является также и R_2-автоматом. А значит, если бы алгоритм существовал для R_2-автоматов, то он же подошел бы и для любого R_1-автомата. □

Лемма 2. *Рассмотрим замкнутые классы $R_1 \subset R_2 \subset \mathbb{P}_2$. Если существует алгоритм определения порядка для R_2-автоматов, то он же подходит для определения порядка R_1-автоматов.*

Доказательство. Очевидно, что любой R_1-автомат является также и R_2-автоматом. Следовательно, алгоритм для R_2-автоматов подойдет и для R_1-автоматов. □

Для замкнутого класса $R \subset \mathbb{P}_2$ мы обозначаем через R^* двойственный ему класс – класс, состоящий из двойственных функций. Для R-автомата G мы обозначаем через G^* двойственный ему автомат, у которого все функции переходов и выходов заменены на двойственные.

Лемма 3. *Рассмотрим замкнутый класс $R \subset \mathbb{P}_2$. Алгоритм определения порядка для R-автоматов существует тогда и только тогда, когда он существует для R^*-автоматов.*

Доказательство. Рассмотрим произвольный автомат $G \in R$. По свойствам двойственности

$$(G^n)^* = (G^*)^n,$$

$$G^n = G^k \Leftrightarrow (G^n)^* = (G^k)^* \Leftrightarrow (G^*)^n = (G^*)^k.$$

То есть, порядки автоматов G и G^* совпадают. Следовательно, для определения порядка одного из них достаточно определить порядок другого. И алгоритм, подходящий для R-автоматов, можно применить к R^*-автоматам, если рассматривать двойственные к ним автоматы того же порядка.

□

Заметим, что мы сознательно требуем в определении R-автомата того, чтобы входной-выходной алфавит Σ и множество состояний Q были целыми булевыми кубами, а не подмножествами булевых кубов. Это сужает класс рассматриваемых автоматов, что усиливает результаты об алгоритмической неразрешимости, но ослабляет результаты о разрешимости. Мотивацией для такого решения стала неудача в попытке обнаружить алгоритм определения порядка для S_6- и P_6-автоматов, у которых входной-выходной алфавит является собственным подмножеством булевого куба. Автор полагает, что такой алгоритм существует, и надеется, что в будущем результат удастся обобщить на более широкий класс автоматов.

3. Доказательство теоремы о классификации

Данный раздел посвящен доказательству основного результата работы, а именно теоремы 1 о классификации классов Поста по разрешимости задачи определения порядка для соответствующих автоматов.

Доказательство. Фактически мы хотим показать, что имеет место следующее разбиение решетки Поста (рис. 1). По леммам 1, 2 для этого достаточно доказать алгоритмическую неразрешимость задачи для F_2^∞-,F_6^∞-,D_2-автоматов и разрешимость для S_6-,L_1-,P_6-автоматов.

Для доказательства неразрешимости задачи мы покажем, что любой автомат может быть изоморфно вложен в F_2^∞-, F_6^∞-, D_2-автоматы, функционирующие схожим образом. Тогда неразрешимость будет следовать из неразрешимости в общем случае [1].

Неразрешимость для D_2-автоматов. Класс D_2 является пересечением монотонных и самодвойственных функций. Рассмотрим произвольный конечный автомат G. Закодируем его входной-выходной алфавит Σ наборами вида $0...010...0 \in \Sigma'$, где длина набора из Σ' не меньше трех. Закодируем множество состояний Q наборами вида $10...010...0 \in \{1\} \times Q'$, где длина набора из Q' не меньше трех, больше длины набора из Σ', и длина набора из $\Sigma' \times \{1\} \times Q'$ нечетная.

Доопределим функции выходов и переходов на остальных наборах так, чтобы они были монотонными и самодвойственными. Без ограничения общности рассмотрим функцию ϕ_1, определяющую значение первого элемента набора, кодирующего состояние. Для каждого набора вида $x1q,\ x \in \Sigma', q \in Q'$ эта функция задает значение $\phi_1(x1q) \in E_2$. Для самодвойственности мы должны доопределить ее на противоположных наборах следующим образом: $\phi_1(\bar{x}0\bar{q}) = \overline{\phi_1(x1q)}$. Для монотонности доопределяем функцию ϕ_1 на некоторых сравнимых с $x1q$ или $\bar{x}0\bar{q}$ наборах: если значение функции равно 0, то на всех меньших наборах тоже полагаем ее равной 0; если значение равно 1, то на всех больших наборах тоже полагаем ее равной 1. Процесс однозначен, так как все наборы $x1q, \bar{x}0\bar{q}$ попарно несравнимы. В самом деле, иначе существовали бы такие наборы $x, x' \in \Sigma',\ q, q' \in Q'$, что $x'1q' > \bar{x}0\bar{q}$, но наборы x, x', q, q' содержат по одной единице и $|x|, |q| > 2$ (по условию кодировки). А значит, наборы $\bar{x}, \bar{q}$ содержат как минимум по две единицы и неравенство $x'1q' > \bar{x}0\bar{q}$ невозможно. Осталось доопределить функцию ϕ_1 на всех остальных наборах. Положим ее равной 0 на всех оставшихся наборах, где число единиц меньше, чем нулей. Положим ее равной 1 на всех оставшихся наборах, где число единиц больше, чем нулей (здесь мы пользуемся нечетностью длины наборов, кодирующих пару из входной буквы и состояния). Очевидно, что такое определение сохраняет монотонность и самодвойственность. Аналогично доопределяем остальные функции ϕ_i, ψ_j.

Полученный D_2-автомат в точности изоморфен автомату G на множестве входных букв Σ'. На остальных буквах из $E^n \setminus \Sigma'$ автомат либо функционирует так же, как на сравнимых наборах из Σ', либо переходит в состояние из одних нулей или одних единиц и становится константным. Действительно, если значение функции ϕ_1 на наборе xlq, где $l \in E_2$, определялось количеством единиц в xlq, то и для всех остальных функций ϕ_i, ψ_j оно будет определяться так же: $\forall i, j, \phi_1(xlq) = \phi_i(xlq) = \psi_j(xlq)$. То есть, входная буква x переводит автомат в состояние из одних нулей или одних единиц, а на выходе мы тоже получаем набор из одних нулей или

одних единиц. Если мы рассматриваем суперпозицию исходного автомата с самим собой несколько раз, то этот набор из одних нулей или одних единиц будет распространяться по всем копиям исходного автомата и переводить их в такие же состояния. Так как длина кодировки состояния больше длины кодировки входной буквы, попавший в состояние из одних нулей или одних единиц автомат никогда из него не выйдет и будет константным. То есть, порядок полученного D_2-автомата совпадает с порядком исходного автомата G.

Неразрешимость для F_2^∞-, F_6^∞-автоматов. Класс F_6^∞ состоит из монотонных функций f, обладающих следующими свойствами: $f(a,...,a) = a$ и все наборы x, на которых $f(x) = 1$, имеют общую единицу. Рассмотрим произвольный конечный автомат G. Закодируем его входной-выходной алфавит Σ наборами вида $0...010...0 \in \Sigma'$. Закодируем множество состояний Q наборами вида $10...010...0 \in \{1\} \times Q'$.

Теперь нам нужно доопределить функции переходов и выходов на остальных наборах. Без ограничения общности рассмотрим функцию ϕ_1, определяющую значение первого элемента набора. Для монотонности, доопределяем функцию ϕ_1 на некоторых сравнимых с $x1q, x \in \Sigma', q \in Q'$ наборах: если значение функции равно 0, то на всех меньших наборах тоже полагаем ее равной 0; если значение равно 1, то на всех больших наборах тоже полагаем ее равной 1. На всех оставшихся наборах полагаем ϕ_1 равной нулю. Очевидно, полученная функция ϕ_1 монотонна, обладает α-свойством ($\phi_1(a,...,a) = a$) и все наборы, на которых она обращается в единицу, имеют общую единицу (в начале кодировки состояния). Аналогично доопределяем остальные функции ϕ_i, ψ_j.

Полученный F_6^∞-автомат изоморфен автомату G на множестве входных букв Σ'. На остальных буквах из $E^n \setminus \Sigma'$ автомат либо функционирует так же, как на сравнимых наборах из Σ', либо переходит в состояние из одних нулей и становится константным, выдавая наборы из нулей. То есть, порядок полученного F_6^∞-автомата совпадает с порядком исходного автомата G. Для F_2^∞-автоматов утверждение следует из леммы 3.

Теперь докажем разрешимость задачи для нижней части решетки.

Разрешимость для L_1-автоматов. Разрешимость для L_1-автоматов следует из разрешимости для линейных автоматов над произвольным конечным полем [2, 4].

Разрешимость для S_6-,P_6-автоматов. Рассмотрим класс S_6-автоматов. Их канонические уравнения содержат только константы и дизъюнкции, а значит, их можно представить в виде

$$\begin{cases} q(t+1) = Aq(t) \vee Bx(t), \\ y(t) = Cq(t) \vee Dx(t), \\ q(0) = q_0, \end{cases}$$

где A, B, C, D - матрицы над E_2, а матричное умножение использует обычное умножение и дизъюнкцию вместо обычного сложения. При таком подходе мы рассматриваем булев куб E_2 как полукольцо (кольцо без вычитания), множество состояний и алфавит как полумодули (как модули, но над полукольцом вместо кольца), а матрицы A, B, C, D как гомоморфизмы этих полумодулей. Кроме того, сложение в E_2 (и в любых многочленах и рядах над ним) идемпотентно ($a \vee a = a$), то есть E_2 – коммутативный диоид. Этот факт позволяет нам воспользоваться следующей теоремой [8]:

Теорема 2. *Пусть S есть произвольный коммутативный диоид и $A \in S^{p\times p}$. Тогда существуют такие $c \geq 1, N \in \mathbb{N}$, что для любых $1 \leq i, j \leq p$ и любого $l \in \{0, ..., c-1\}$ найдется конечное семейство скаляров $\alpha_1, \lambda_1, ..., \alpha_k, \lambda_k$, такое, что*

$$\forall n \geq N,\ A_{ij}^{nc+l} = \bigoplus_{r=1}^{k} \alpha_r \lambda_r^{n-N}.$$

Аналогично случаю линейных автоматов, мы можем описать действие S_6-автомата G с помощью передаточной функции $M(z) = \bigvee_{v=0}^{\infty} CA^v Bz^{v+1} \vee D$ и сдвига $S(z) = \bigvee_{v=0}^{\infty} CA^v q_0 z^v$.

$$y(z) = M(z)x(z) \vee S(z).$$

$M(z)$ – это матрица над коммутативным диоидом, а потому к ней применима теорема 2 и

$$\forall t \geq N,\ M_{ij}^{tc+l}(z) = \bigvee_{r=1}^{k} \alpha_r(z) \lambda_r^{t-N}(z).$$

Заметим, что, для произвольного $l \in \{0, ..., c-1\}$ верно

$$\#\{M_{ij}^{tc+l}\}_{t\geq N} < \infty \ \Leftrightarrow\ \#\{M_{ij}^{t}\}_{t\geq 0} < \infty,$$

а значит, для определения порядка передаточной функции нам достаточно рассмотреть одно произвольное значение l. Для фиксированных i, j, l возможны следующие случаи:

1) $\alpha_r(z) = 0$ или $\lambda_r(z) = 0$. Очевидно, $\mathrm{ord}(\alpha_r(z)\lambda_r^{t-N}(z)) < \infty$.

2) $\langle \lambda_r(z), z^0 \rangle = 0$. Тогда минимальная степень z в $\alpha_r(z)\lambda_r^{t-N}(z)$ будет расти с ростом t и $\mathrm{ord}(\alpha_r(z)\lambda_r^{t-N}(z)) = \infty$.

3) $\langle \lambda_r(z), z^0 \rangle = 1$ и $\lambda_r(z) \notin E_2[z]$. Получается, $\lambda_r(z)$ бесконечный ряд с единицей. При его возведении в степень возможно лишь добавление новых слагаемых. Так как он периодический, добавление нового слагаемого происходит с добавлением бесконечного числа других слагаемых со

сдвигом равным длине периода ряда. Так как период конечен, то конечно и число возможных рядов при возведении $\lambda_r(z)$ в степень. А значит, $\mathrm{ord}(\alpha_r(z)\lambda_r^{t-N}(z)) < \infty$.

4) $\langle \lambda_r(z), z^0 \rangle = 1$, $\lambda_r(z) \in E_2[z]$ и $\alpha_r(z) \in E_2[z]$. В таком случае $\lambda_r(z)$ и $\alpha_r(z)$ многочлены, и, очевидно, что $\mathrm{ord}(\alpha_r(z)\lambda_r^{t-N}(z)) = \infty$.

5) $\langle \lambda_r(z), z^0 \rangle = 1$, $\lambda_r(z) \in E_2[z]$ и $\alpha_r(z) \notin E_2[z]$. Аналогично случаю (3), с ростом n возможно лишь добавление новых слагаемых, причем добавляются они сразу со всеми сдвигами, кратными длине периода. Следовательно, $\mathrm{ord}(\alpha_r(z)\lambda_r^{t-N}(z)) < \infty$.

Ясно, что мы можем определить порядок и конечной суммы таких рядов $M_{ij}^{tc+l}(z) = \bigvee_{r=1}^{k} \alpha_r(z)\lambda_r^{t-N}(z)$. Таким образом, мы умеем определять порядок каждого элемента матрицы $M(z)$. Если порядки всех элементов передаточной функции конечны, то есть конечен порядок самой передаточной функции, то очевидно, конечен и порядок всего автомата. Также порядок конечен, если начиная с какого-то момента все коэффициенты в сдвиге $S(z)$ равны единицам.

Пусть существует элемент $S_i(z)$ в векторе $S(z)$ с нулевым коэффициентом при бесконечном числе мономов z^k. Рассмотрим соответствующую строку $M_i(z)$ матрицы $M(z)$. Если в ней все элементы имеют конечный порядок, то, очевидно, порядок автомата по этой координате тоже конечен. Если же имеется хотя бы один элемент M_{ij} бесконечного порядка, то при возведении матрицы в степень у него будет расти либо максимальная, либо минимальная степень. Рассмотрим входной вектор $x(z)$, такой что $x_j(z) = z^q$ для произвольного q, а для всех остальных координат l он равен нулю $x_l(z) = 0$. Если периоды в $S_i(z)$ и $M_{ij}^N(z)$ не совпадают, то порядок автомата на входе $x(z)$ бесконечный. Если периоды совпадают, то нужно выбрать число q в $x_j(z) = z^q$ так, чтобы минимальная или максимальная степень в $M_{ij}^t(z)x_j(z)$ всегда соответствовала нулевому сдвигу в $S_i(z)$. В таком случае порядок автомата на входе $x(z)$ тоже будет бесконечным. А значит, будет бесконечным и порядок самого автомата. Повторяем процедуру для всех координат. Получили алгоритм проверки конечности порядка автомата. Для P_6-автоматов утверждение следует из леммы 3. Теорема доказана.

□

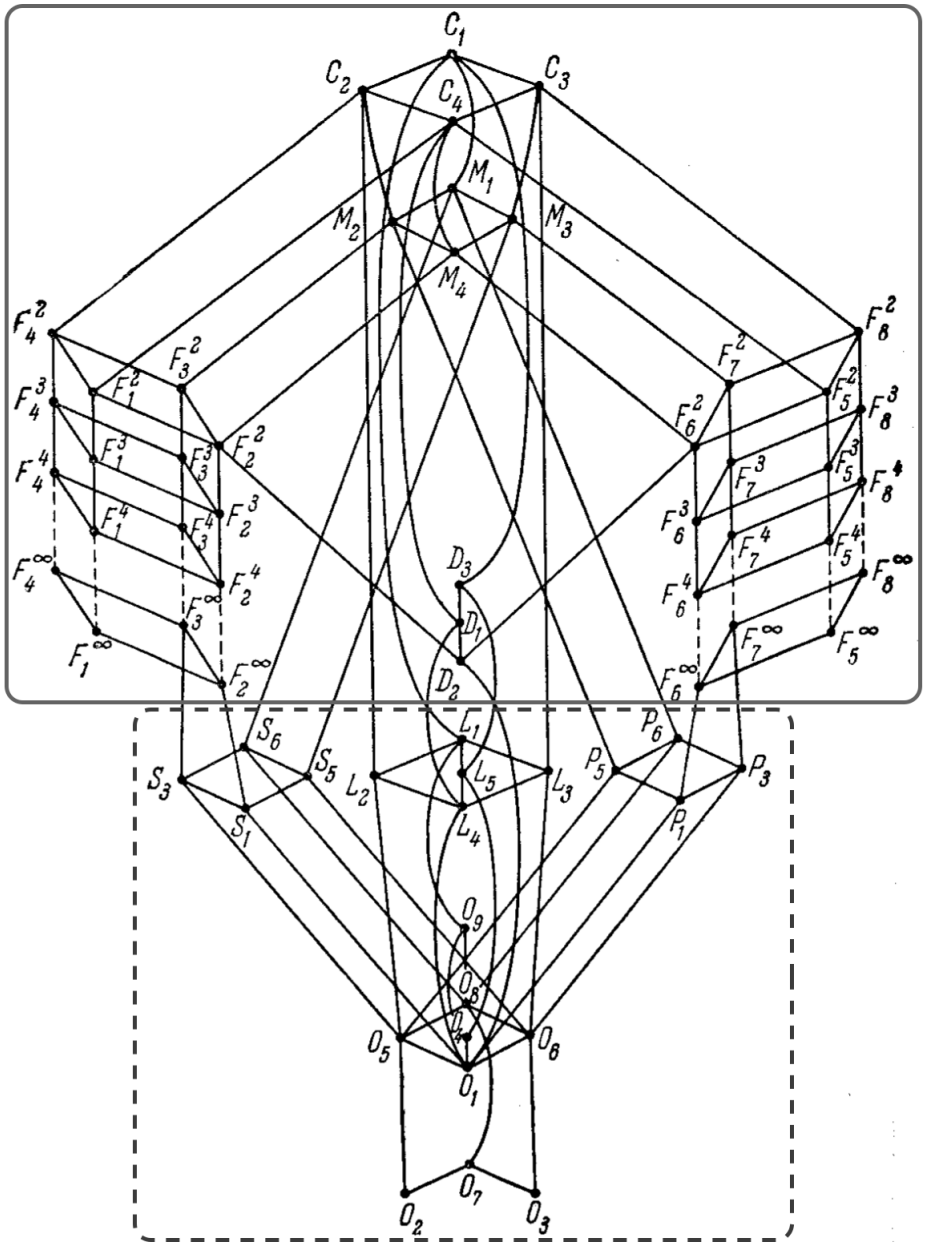

Рис. 1. Решетка Поста замкнутых классов $\mathbb{P}_2$ (схема взята из [9]). Сплошным прямоугольником выделены классы, для которых алгоритмически неразрешима задача определения порядка в соответствующем классе автоматов. Пунктирным прямоугольником выделены классы, порождающие классы автоматов с разрешимой задачей определения порядка.

Список литературы

[1] P. Gillibert, “An automaton group with undecidable order and Engel problems”, *Journal of Algebra*, **497** (2018), 363–392.

[2] Муравьев Н.В., “Разрешимость задачи определения порядка линейного автомата”, *Интеллектуальные системы. Теория и приложения*, **24**:2 (2020), 145–155.

[3] Муравьев Н.В., “О порядках линейных над полем рациональных чисел автоматов”, *Интеллектуальные системы. Теория и приложения*, **24**:4 (2020), 119–124.

[4] Муравьев Н.В., “Оценки порядков линейных автоматов”, *Вестник Московского университета. Серия 1: Математика, Механика*, 2022, № 6, 8–14.

[5] Муравьев Н.В., “Максимальные конечные порядки линейных автоматов над произвольным полем”, *Вестник Московского университета. Серия 1: Математика, Механика*, 2024, № 5, 71–73.

[6] Бабин Д.Н., “О классификации автоматных базисов Поста по разрешимости свойств полноты и A-полноты”, *Доклады академии наук*, **367**:4 (1999), 439–441.

[7] Бабин Д.Н., “Разрешимость задачи полноты автоматного базиса в зависимости от его булевой части”, *Вестник Московского университета. Серия 1: Математика, Механика*, 2019, № 1, 52–54.

[8] Stéphane Gaubert, “On rational series in one variable over certain dioids. [Research Report] RR-2162, inria-00074510 INRIA”, 1994.

[9] Яблонский С.В., Гаврилов Г.П., Кудрявцев В.Б., *Функции алгебры логики и классы Поста*, "Наука", 1966.

The order problem for automata which transition and output functions lie in closed Post classes

Muravev N.V.

We consider the order problem for Mealy automata with respect to the superposition operation.

The splitting of the Post's lattice of closed classes is proved based on decidability of the order problem for respective R-automata.

Keywords: Mealy automata, Post classes, algorithmic decidability.

References

[1] P. Gillibert, “An automaton group with undecidable order and Engel problems”, *Journal of Algebra*, **497** (2018), 363–392.

[2] N.V. Muravev, “Decidability of the order problem for linear automata”, *Intelligent systems. Theory and applications*, **24**:2 (2020), 145–155.

[3] N.V. Muravev, “About orders of linear over rationals automata”, *Intelligent systems. Theory and applications*, **24**:4 (2020), 119–124.

[4] N.V. Muravev, “Bounds on Orders of Linear Automata”, *Moscow University Mathematics Bulletin*, 2022, № 6, 8–14.

[5] N.V. Muravev, “Maximal Finite Orders of Linear Automata over an Arbitrary Field”, *Moscow University Mathematics Bulletin*, 2024, № 5, 71–73.

[6] D.N. Babin, “On classification of automata Post basises based on decidability of completeness and A-completeness”, *Reports of the Academy of Science*, **367**:4 (1999), 439–441.

[7] D.N. Babin, “Solvability of the Problem of Completeness of Automaton Basis Depending on its Boolean Part”, *Moscow University Mathematics Bulletin*, 2019, № 1, 52–54.

[8] Stéphane Gaubert, “On rational series in one variable over certain dioids. [Research Report] RR-2162, inria-00074510 INRIA”, 1994.

[9] Yablonsky S.V., Gavrilov G.P., Kudryavtsev V.B., *Functions of the logic algebra and Post classes*, "Nauka", 1966.

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://intsysmagazine.ru/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), для аспирантов ФИО научного руководителя, 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. Список на русском языке приводится в конце файла с текстом статьи, в то время как список, переведённый на английский язык, прилагается отдельным файлом.
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. Авторам бесплатно предоставляется номер журнала, в котором вышла статья. 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://intsysmagazine.ru,

и доступ к ним бесплатный. Там же будут размещены полные тексты всех публикуемых статей.

Подписано в печать: 25.09.2025
Дата выхода: 01.10.2025
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).