

УДК 004.415.2:519.681

 10.25209/2079-3316-2025-16-2-3-54

Сложность вычислений с путешествиями во времени

Милад Джудакизаде¹, Анатолий Петрович Бельтюков^{2✉}^{1,2} Удмуртский государственный университет, Ижевск, Россия^{2✉} belt.udsu@mail.ru

Аннотация. Эта работа рассматривает математическую модель вычислений, которая может интерпретироваться как компьютер, способный получать данные из будущих состояний своего вычислительного процесса. При определённых сочетаниях входных данных и программ такие вычисления могут становиться невыполнимыми из-за возникающих противоречий или приводить к неоднозначным результатам. Мы исследуем программы, для которых этот процесс всегда выполним и даёт однозначный результат.

Показано, что при отсутствии ограничений на вычислительную сложность такие машины могут выдавать значение любого рекурсивно разрешимого предиката через фиксированное время после начала вычисления – время задержки ответа (при этом процесс вычисления должен продолжаться и после получения ответа). Если общее время работы таких машин ограничить полиномами от размера входных данных, то эти машины будут распознавать в точности языки, принадлежащие пересечению классов NP и co-NP, за постоянное время задержки ответа в таком же смысле.

Рассматриваются возможные реализации такого компьютера на практике, включая анализ возможных протоколов работы с использованием квантового отжига для выбора нужного процесса. Показано, что при распараллеливании вычислительного процесса класс задач, решаемых рассматриваемыми машинами за полиномиальное время, соответствует классу PSPACE.

Кроме того, исследуется режим работы, при котором эти машины имеют прямой доступ ко входным данным. В этом случае, если время работы ограничено логарифмом от размера входных данных, то класс задач, решаемых таким параллельно работающим компьютером, содержит LOGSPACE.

Результаты данного исследования могут быть использованы для разработки новых принципов программирования недетерминированных вычислительных машин, в которых вместо недетерминированных выборов используется передача данных из будущего.

Ключевые слова и фразы: Машина Тьюринга, вычислительная сложность, рекурсивные множества, недетерминированные вычисления, квантовый отжиг, машина времени, искусственный интеллект

Для цитирования: Джудакизаде М., Бельтюков А. П. *Сложность вычислений с путешествиями во времени* // Программные системы: теория и приложения. 2025. Т. 16. № 2(65). С. 3–54. https://psta.psiras.ru/read/psta2025_2_3-54.pdf

Введение

В данной статье мы рассматриваем необычный взгляд на вычислительные процессы. Что получается, если машине разрешать заглядывать в своё собственное будущее и использовать эту информацию для принятия решений в настоящем? Эта идея, которая на первый взгляд кажется внутренне противоречивой, лежит в основе исследуемой вычислительной модели — машины Тьюринга с каналом передачи данных против течения времени. Эта модель не только переосмысливает понимание времени в вычислениях, но и поднимает фундаментальные вопросы о причинности и взаимодействии между прошлым и будущим в вычислительных процессах.

Наша машина определена таким образом, что она может получать информацию из будущих состояний своего вычислительного процесса. Эта модель представляет собой теоретическую конструкцию, вводящую новые способы организации вычислений. Она позволяет получить результат сложного вычисления до его завершения и использовать эту информацию. Как и в любом путешествии во времени, здесь возникают сложности: передача данных из будущего может привести к противоречиям и неоднозначностям.

В данной статье мы сосредотачиваемся на случаях, когда таких проблем можно избежать, и демонстрируем, как эту модель можно реализовать и применить для решения сложных задач. Этот подход представляет собой альтернативную вычислительную парадигму, которая связывает время, причинность и вычисления, позволяя применять новые стратегии решения задач.

Одним из результатов является то, что такая машина может вычислить значение любого рекурсивно разрешимого предиката за постоянное время. Это означает, что задачи, требующие значительного времени в классических моделях, могут быть решены практически мгновенно (хотя сам вычислительный процесс должен продолжаться и общее его время может оказаться весьма значительным).

Мы доказываем также, что если общее время работы таких машин ограничивать полиномами от размера входных данных, то эти машины смогут распознавать все языки из пересечения классов NP и co-NP. Эти результаты говорят о том, что предложенная модель не только теоретически интересна, но и обладает значительной вычислительной мощностью.

Мы также рассматриваем модификацию машины, которая позволяет осуществлять параллельную обработку данных. Показано, что в этом случае класс задач, решаемых такими машинами за полиномиальное

время, соответствует PSPACE (то есть полиномиальное время на новых машинах эквивалентно PSPACE на обычных машинах Тьюринга).

Другая модификация предполагает добавление механизма прямого доступа ко входным данным через адресацию ячеек входной ленты с помощью специальной адресной ленты. В этом случае, при ограничении времени работы логарифмом от размера входных данных, класс задач, решаемых нашим распараллеленным компьютером, включает LOGSPACE (здесь LOGTIME также преобразуется в соответствующую емкостную сложность). Стоит отметить, что такой режим работы не влечёт чрезмерно высокой сложности и является реализуемым на обычных компьютерах при разумных размерах входных данных.

С практической точки зрения, наша работа вводит новые механизмы программирования. Вместо сложных недетерминированных выборов, которые могут быть трудны для понимания, мы предлагаем использовать данные из будущего, что делает процесс вычислений интуитивно более понятным. Мы также исследуем возможные реализации этой модели, включая детерминированный поиск с возвратом и квантовый отжиг.

В первом подходе систематический поиск с возвратами назад гарантирует, что все возможные пути будут исследованы для предотвращения парадоксов. Во втором подходе квантовые адиабатические машины одновременно исследуют множество состояний вычислительного процесса, выбирая самосогласованную последовательность действий.

Таким образом, мы предлагаем модель, которая расширяет наше понимание вычислительных процессов и предоставляет новые подходы к программированию.

1. Связанные работы

За последние несколько десятилетий идея использования временных парадоксов и передачи данных во времени привлекла внимание в теоретической информатике и физике. Многие исследователи изучали возможность расширения классических вычислительных моделей за счёт включения механизмов, позволяющих получать информацию из будущего, что, как предполагается, может переопределить традиционные взгляды на вычислительные ресурсы и сложность.

В основополагающей работе Кука [1] было введено понятие NP-полноты, заложившее основу для дальнейших исследований в области вычислительной сложности и сводимости различных задач к универсальной проблеме. Развивая эти идеи, Р. М. Карп [2] проанализировал сводимость комбинаторных задач, показав, что многие на первый взгляд различные задачи взаимосвязаны.

Классическая работа Гэри и Джонсона [3] стала фундаментальным справочником для исследователей, изучающих вычислительную сложность, представив широкий спектр NP-полных задач и послужив отправной точкой для анализа ограничений традиционных вычислительных моделей.

В последние годы значительное внимание уделяется моделям, использующим физические эффекты для решения сложных вычислительных задач. Ааронсон [4] исследовал потенциал квантовых вычислений для решения традиционно сложных задач, продемонстрировав, что квантовые методы могут предоставить существенные преимущества перед классическими алгоритмами.

Дойч [5] ввёл концепцию универсального квантового компьютера, что стало важной вехой в квантовых вычислениях и повлияло на дальнейшие исследования гибридных вычислительных моделей.

Что касается моделей, нарушающих классическую причинность, существенный вклад был сделан в исследованиях, связанных с замкнутыми времениподобными кривыми (ЗВК). Брун [6] показал, что использование ЗВК может значительно ускорить решение задач. Кроме того, Ааронсон и Уотрус [7] установили эквивалентность квантовых и классических вычислений в присутствии ЗВК, подчеркнув потенциал таких моделей для расширения вычислительных возможностей.

Помимо классических подходов, идея интеграции передачи данных на основе путешествий во времени с концепциями теории сложности вызвала интерес. Работы Пападимитриу [8] и Ароры и Барака [9] предоставляют глубокий анализ структуры вычислительных задач, включая связь между классами NP и co-NP. Эти исследования служат основой для анализа того, как нестандартные вычислительные модели, использующие передачу данных из будущего, могут решать задачи, с которыми традиционные модели справляются с трудом.

Классические работы по квантовым вычислениям, такие как книга Нильсена и Чанга [10], а также алгоритмы Гровера [11] и Шора [12], иллюстрируют, как квантовые методы могут значительно ускорить решение определённых классов задач.

Более поздние исследования, такие как работы Харроу и Монтанаро [13] и исследования Икбала и др. [14], подтверждают практическую значимость квантовых алгоритмов для решения сложных вычислительных задач.

Важным аспектом этой области является концепция передачи данных на основе путешествий во времени, позволяющая машинам предсказывать свои будущие состояния и использовать эту информацию для корректировки текущих вычислений. В этом контексте работа Орлицки [15] исследует

модель машин с путешествиями во времени и рассматривает проблему самосогласованной остановки. Этот подход расширяет традиционные модели, предлагая новые взгляды на взаимодействие временных потоков в вычислениях.

Кроме того, вклад Гёделя [16] продемонстрировал теоретическую возможность замкнутых времениподобных кривых в космологических решениях, предоставив дополнительное физическое обоснование для вычислительных моделей, включающих передачу данных из будущего.

Исследования Бэкона [17] и Вольперта [18] указывают на то, что физические ограничения на предсказание будущего и обратную связь между временными состояниями значительно влияют на вычислительные процессы.

Наконец, экспериментальные исследования, такие как работа Кинга и др. [19], показывают, что квантовые системы с тысячами кубитов уже могут моделировать сложные физические явления, что вселяет надежды на практическую реализуемость вычислительных моделей, использующих эффекты путешествий во времени.

Таким образом, эти работы иллюстрируют, что идея передачи данных из будущего и эффектов временных петель в вычислениях имеет прочные теоретические и практические основания. Наша модель вычислений с путешествиями во времени естественным образом вписывается в этот контекст, предлагая новый взгляд на вычислительную сложность и позволяя решать задачи, недоступные для традиционных моделей.

2. Структура и особенности машины Тьюринга с каналом передачи данных во времени

В данном разделе описывается конструкция машины Тьюринга, оснащённой каналом передачи данных во времени. Эта модель расширяет классическую машину Тьюринга за счёт включения механизма, позволяющего передавать данные на ранее выполненные шаги вычислительного процесса и получать данные, переданные из будущих шагов. Рассмотрим структуру и особенности такой машины.

2.1. Основные компоненты машины

Машина Тьюринга с каналом передачи данных во времени является многоленточной машиной (с одной головкой на каждую ленту) и включает следующие элементы:

Входная лента: На этой ленте содержится входное слово, которое необходимо обработать. В начале работы машины входные данные записываются на эту ленту.

Выходная лента: На этой ленте записывается результат работы машины. Для распознающей машины результат может быть представлен в виде бинарного значения: 0 (нет) или 1 (да).

Несколько рабочих лент для выполнения необходимых вычислений.

Коммуникационная лента (лента связи): Специальная лента, предназначенная для передачи данных во времени. Машина может записывать данные на эту ленту для отправки в прошлое и получать данные из будущего. Для отправки и получения данных машина должна перейти в специальные состояния передачи и приёма данных соответственно.

Адресная лента для передаваемых данных: На этой ленте записываются метки, которые позволяют извлекать данные, ссылаясь на эти метки в запросах. Для передачи и извлечения данных используются специальные состояния машины.

В состоянии передачи содержимое коммуникационной ленты вместе с меткой отправляется в канал передачи данных во времени. Если данные с такой меткой уже были ранее отправлены, то считаем, что при работе машины возникла ошибка.

В состоянии приёма коммуникационная лента получает данные из канала передачи, связанные с заданной меткой. Если таких данных в будущем не оказалось, то также считаем, что при работе машины возникла ошибка. Для решения задач распознавания машины, при работе которых могут возникать ошибки, использовать запретим.

Алфавиты лент: Каждая лента имеет свой алфавит символов, которые могут быть записаны на неё.

Управляющее устройство: На каждом шаге работы управляющее устройство может находиться в одном из состояний алфавита состояний. Выделяются следующие специальные состояния:

начальное состояние, в котором машина начинает вычислительный процесс;

заключительное состояние, в котором работа машины завершается;

состояние передачи данных, при переходе в которое данные передаются с указанной меткой в канал передачи данных;

состояние приёма данных, при переходе в которое данные принимаются из канала передачи данных с указанной меткой;

состояние вывода, при переходе в которое содержимое выходной ленты отправляется во внешний мир, пока машина продолжает свою работу; считаем, что машина работает с однократной записью на выходную ленту, то есть повторный переход в состояние вывода — ошибка в работе машины.

Программа: Функция (таблица), которая отображает состояние машины и символы, видимые головками на каждой ленте, в новое состояние, новые символы, записываемые на ленты, и направления сдвигов головок.

Время вычислений: Последовательность шагов. Каждый шаг выполняет одну команду машины или передаёт/принимает один символ через канал передачи данных во времени (то есть на передачу длинного слова в канал передачи данных тратится столько шагов, какова длина этого слова). Для распознающей машины время вычислений должно быть конечным.

На диаграмме рисунка 1 показан процесс обработки данных с возможностью передачи информации из будущего в прошлое через временной сдвиг, реализуемый замкнутыми времениподобными кривыми в метрике Керра.

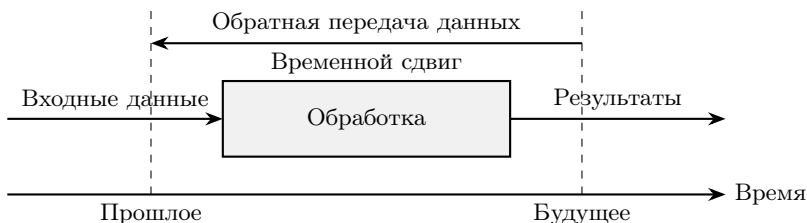


Рисунок 1. Схема работы машины Тьюринга с обратной временной связью

2.2. Особенности работы машины

Работа машины Тьюринга с каналом передачи данных во времени имеет несколько отличительных особенностей:

Вывод результата до завершения вычислений: Машина может вывести результат на выходную ленту до завершения процесса вычислений. Это приводит к разделению двух понятий времени:

время задержки ответа,
общее время вычислений.

Однозначность и определённости вычислений: Из-за возможности получения данных из будущего работа машины может стать неоднозначной или неопределённой. Например, неоднозначность возникает, если машина, получив некоторые данные из будущего, записывает те же данные с той же меткой, что может привести к недетерминированности процесса и, как следствие, к различным конечным результатам, в зависимости от того, какие данные окажутся прочитанными и записанными. Если машина, прочитав данные с одной меткой, затем обязательно записывает отличные от прочитанных данные с той же меткой, то возникает противоречие, делающее процесс невозможным, что приводит к неопределённости. Машина считается

распознающей некоторое свойство входных данных (т.е. вычисляющей характеристическую функцию некоторого множества), если её работа определена для всех входных данных и результат однозначен.

Программируемость: В конце статьи мы опишем инструменты алгоритмических языков, которые позволят программировать подобные действия не только для машин Тьюринга, но и для других компьютеров.

2.3. Временные аспекты работы машины

Ключевой особенностью машины является возможность получения данных из будущего, что приводит к следующим временным эффектам:

Постоянное время задержки ответа: Благодаря механизму передачи данных во времени машина может вывести простой результат (ДА или НЕТ) за постоянное время, независимо от вычислительной сложности.

Общее время вычислений: Несмотря на постоянное время задержки ответа, общее время вычислений может быть значительно больше, так как вычислительный процесс должен продолжаться даже после вывода результата.

2.4. Объяснение неоднозначности и неопределённости

Чтобы проиллюстрировать неоднозначность и неопределённость в работе машины, рассмотрим следующий пример. Предположим, что машина запрашивает один бит из будущего и выбирает ветвь вычислений на основе его значения. Если получено значение бита 1, то машина следует по первой ветви; если получено значение 0, то она следует по второй ветви. Неоднозначность процесса вычисления возникает, если в каждой ветви машина записывает тот же бит с той же меткой на коммуникационную ленту. Это создаёт два возможных пути вычисления. Следует отличать неоднозначность вычисления от неоднозначности результата. При неоднозначном процессе вычисления результат может оказаться одним и тем же.

Если машина считывает один бит из будущего и пытается записать противоречащий ему бит на коммуникационную ленту с той же меткой, то возникает противоречие, приводящее к невозможности такого вычисления.

Комбинации этих ситуаций могут приводить к тому, что некоторые ветви процесса вычисления, возникшие из-за внутренней неоднозначности, затем могут оказаться противоречивыми и быть по этой причине отсеяны. Тогда останутся только непротиворечивые ветви. Может оказаться так, что все оставшиеся непротиворечивые ветви приводят к одному окончательному результату. Если всегда происходит так, то будем считать такую машину корректной для решения задачи распознавания.

Таким образом, описанные ситуации могут привести к следующему:

- к неоднозначности*, если несколько ветвей вычислений завершаются разными результатами,
- к неопределённости*, если ни одна ветвь не может завершиться без противоречий.

Это демонстрирует, как использование данных из будущего может повлиять на вычислительный процесс.

3. Возможности машин с передачей данных во времени

Схема передачи данных во времени в машине Тьюринга с замкнутыми времениподобными кривыми представлена на рисунке 2. Показаны основные этапы обработки информации, механизм обратной связи через временные петли и направление передачи данных.

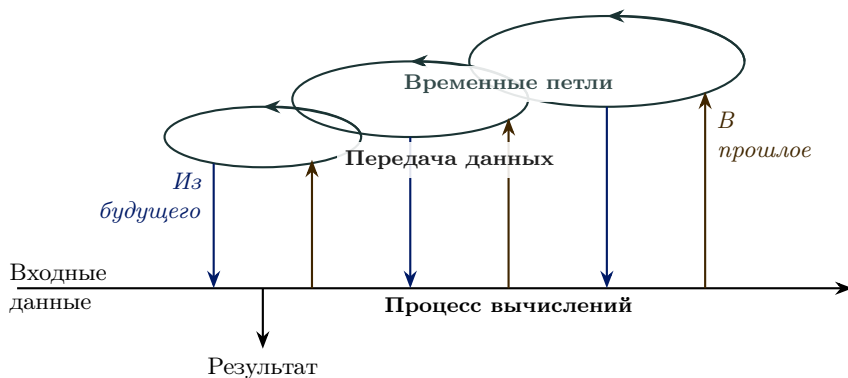


Рисунок 2. Схема временной передачи данных в машине Тьюринга с замкнутыми времениподобными кривыми

В данном разделе рассматриваются вычислительные возможности машины Тьюринга с каналом передачи данных во времени. Мы представляем две теоремы, которые демонстрируют, как использование данных из будущего влияет на вычислительную сложность и классы задач, решаемых такой машиной. Эти теоремы показывают, что предложенная модель обладает свойствами, позволяющими решать задачи, недоступные для классических вычислительных систем.

3.1. Распознавание разрешимых множеств

ТЕОРЕМА 1. Машина Тьюринга с каналом передачи данных во времени распознаёт с постоянным временем задержки ответа в точности все рекурсивно разрешимые множества.

ДОКАЗАТЕЛЬСТВО. Доказательство основывается на способности машины использовать данные из будущего для предсказания результатов вычислений. Пусть нам надо распознать принадлежность входных слов к заданному рекурсивно разрешимому множеству. Для этого множества имеется обычная распознающая машина Тьюринга. Из этой машины построим требуемую машину Тьюринга с каналом передачи данных во времени, которая будет работать следующим образом.

В начале работы машина запрашивает содержимое коммуникационной ленты с меткой 0. Это значение и выдаётся на выход в качестве результата работы. Затем запускается процесс работы обычной машины Тьюринга, распознающей принадлежность заданному множеству. Когда распознавание закончится, его результат передаётся в прошлое с меткой 0. Таким образом обеспечивается, что в прошлом машина выдала верный результат.

Теперь пусть мы имеем исходную машину с каналом передачи данных по времени, распознающую принадлежность входных слов некоторому множеству. По определению, общее время работы такой машины всегда конечно. Построим новую обычную машину Тьюринга, распознающую принадлежность этому множеству, которая перебирает все возможные слова, претендующие на роль частичного протокола работы исходной машины (цепочки мгновенных описаний), в порядке их удлинения.

Машина проверяет для каждого слова, является ли оно правильным завершённым протоколом работы исходной машины. Как только такой протокол найден, результат работы исходной машины, выдаётся в качестве результата новой машины. Поскольку процесс работы исходной машины, по определению, определён и однозначен, новая машина всегда выдаёт правильный ответ о принадлежности входного слова заданному множеству.

□

3.2. Распознавание множеств в $NP \cap co-NP$

ТЕОРЕМА 2. Машина Тьюринга с постоянным временем задержки ответа и полиномиальным общим временем вычислений распознаёт в точности все множества в $NP \cap co-NP$.

ИДЕЯ ДОКАЗАТЕЛЬСТВА. интерпретируя приём данных из будущего как ветвление на несколько путей, мы строим машину, которая работает за полиномиальное время и обеспечивает согласованность результатов на всех ветвях вычислений.

4. Доказательство теоремы 2

В данном разделе мы изложим доказательство теоремы 2, сформулированной в предыдущем разделе. Сначала мы переформулируем работу нашей машины в терминах недетерминированных вычислений. Это позволит нам преобразовать работу нашей машины в работу обычных недетерминированных машин Тьюринга и наоборот. Поскольку механизм перехода к постоянному времени задержки описан при доказательстве предыдущей теоремы, то для начала достаточно доказать теорему 2 без упоминания об этом, что мы и сделаем.

4.1. Переход к недетерминированным вычислениям

Идея заключается в том, чтобы интерпретировать чтение каждого символа данных на коммуникационной ленте, полученного из будущего по некоторой метке, как ветвление процесса вычислений. Каждая ветвь соответствует одному из возможных значений этого символа. Когда машина достигает момента времени, обозначенного данной меткой, она проверяет, совпал ли предполагаемый считанный символ с переданным. Если значение символа, записанного в ячейке коммуникационной ленты, совпадает с ожидаемым значением в текущей ветви, вычисления продолжаются. В противном случае ветвь вычислений завершается из-за возникшего противоречия.

Корректная распознающая машина должна давать одинаковые ответы (либо 0 — НЕТ, либо 1 — ДА) на всех непротиворечиво завершающихся ветвях для данного входного слова. Далее мы преобразуем нашу исходную машину с передачей данных во времени в недетерминированную, как показано выше, а затем разделим её на две недетерминированные машины Тьюринга, обработав, упомянутые выше противоречия следующим образом.

- (1) Первая машина принимает распознаваемое множество, возвращая 1, когда исходная машина выводит 1, а во всех остальных случаях (включая противоречия) возвращает 0.
- (2) Вторая машина принимает дополнение этого множества, возвращая 1, когда исходная машина выводит 0, а во всех остальных случаях (включая противоречия) возвращает 0.

Таким образом, если исходная машина с передачей данных во времени выдаёт да, то первая из полученных машин на некоторых ветвях вычисления выдаёт ДА, а вторая — всегда НЕТ. Наоборот, если исходная машина выдаёт НЕТ, то первая всегда выдаёт НЕТ, а вторая — на некоторых ветвях вычисления выдаёт ДА. Таким образом, первая из построенных детерминированных принимает само распознаваемое множество, а вторая — его дополнение. Нетрудно видеть, что полиномиальное ограничение времени при указанном преобразовании сохраняется, так как все проведённые перестройки дают не более, чем полиномиальное замедление. Так что, если исходная машина распознавала данное множество за полиномиальное время, то такое же время работают и вновь построенные машины. Это означает, что исходно распознаваемое множество принадлежит $NP \cap co-NP$.

4.2. Построение машины с передачей данных во времени из недетерминированных машин

Обратное утверждение также верно: если существуют две недетерминированные машины, ограниченные полиномиальным временем, одна из которых принимает некоторое множество, а другая принимает его дополнение, мы можем из них построить одну машину с передачей данных из будущего, распознающую это множество и работающую полиномиальное время. Эта машина работает следующим образом:

- (1) В начале процесса она угадывает, какую из двух недетерминированных машин запустить.
- (2) Затем она выполняет выбранную машину.
- (3) Если первая выбранная машина выводит 1, ветвь вычислений завершается с результатом 1.
- (4) Если первая выбранная машина выводит 0, ветвь вычислений приводится к противоречию.
- (5) Если вторая выбранная машина выводит 1, ветвь вычислений завершается с результатом 0.

- (6) Если вторая выбранная машина выводит 0, ветвь вычислений приводится к противоречию.

По условию, ровно одна из этих машин должна дать положительный результат. Таким образом, наша машина всегда даёт определённый ответ: либо 1, либо 0.

4.3. Постоянное время задержки ответа

Постоянное время задержки ответа достигается благодаря тому, что решение о результате вычислений принимается в самом начале процесса. Машина запрашивает данные из будущего и использует их для немедленного вывода ответа. При этом процесс вычислений должен продолжаться. $\square$

5. Вопросы возможной реализации

В данном разделе обсуждаются возможные подходы к практической реализации машины Тьюринга с каналом передачи данных во времени. Хотя такая машина является теоретической конструкцией, её можно эмулировать с использованием существующих вычислительных методов и технологий. Основные подходы к реализации включают детерминированный поиск с возвратом, методы отжига, в частности, квантовый отжиг. Каждый из этих методов имеет свои особенности и ограничения, которые обсуждаются ниже.

5.1. Детерминированный поиск с возвратом

Один из самых простых способов реализации машины с передачей данных из будущего заключается в использовании детерминированного поиска с возвратом на обычном компьютере. В этом подходе процесс вычислений моделируется как дерево возможных состояний, где каждая ветвь соответствует одному из возможных значений символа, полученного из будущего.

Принцип работы:

- (1) На каждом шаге вычислений, при чтении символа данных с коммуникационной ленты, машина проверяет все возможные значения этого символа.
- (2) Если выбранное значение приводит к противоречию, система возвращается к точке принятия решения и пробует следующее значение.
- (3) Процесс продолжается до тех пор, пока не будет найдено согласованное решение или не будут исчерпаны все возможные значения.

Преимущества:

- Простота реализации на существующих вычислительных платформах.
- Возможность использования стандартных алгоритмов поиска с возвратом.

Недостатки:

- Высокая вычислительная сложность, особенно для задач с большим количеством ветвлений.
- Экспоненциальный рост времени вычислений с увеличением глубины дерева решений.

5.2. Методы отжига

Методы отжига представляют собой более сложный подход к реализации машины с передачей данных во времени. В этом случае процесс вычислений рассматривается как целостная структура (время разворачивается в пространстве). Затем выполняется поиск состояния этой структуры с наименьшей энергией, где энергия соответствует количеству противоречий в системе.

Принцип работы:

- (1) Изначально система находится в возбуждённом состоянии с высоким уровнем энергии, что соответствует наличию множества противоречий.
- (2) Постепенно система охлаждается, переходя в состояния с меньшей энергией.
- (3) Если в процессе отжига достигается состояние с наименьшей энергией (без противоречий), процесс считается успешным.
- (4) Если наименьшая энергия достигается только для одного ответа (0 или 1), результат считается окончательным.

Преимущества:

- Возможность нахождения глобального минимума энергии даже в сложных системах с множеством локальных минимумов.
- Эффективность для задач, где противоречия могут быть смягчены или устранены в процессе оптимизации.

Недостатки:

- Требуется значительных ресурсов для достижения состояния с минимальной энергией.
- Не гарантирует нахождение оптимального решения за полиномиальное время.

5.3. Квантовый отжиг

Квантовый отжиг является перспективным методом реализации машины с передачей данных из будущего, основанным на использовании квантовых устройств. Этот метод использует эффект квантового туннелирования для нахождения состояний с наименьшей энергией.

Преимущества:

- Потенциально высокая скорость нахождения решений благодаря квантовым эффектам.
- Возможность решения задач, сложных для классических компьютеров.

Недостатки:

- Скорость и вычислительная эффективность квантового отжига ещё не до конца изучены.
- Существующие квантовые устройства ограничены по объёму памяти и уровню шума, что осложняет их использование для сложных задач.

Перспективы: Несмотря на текущие ограничения, квантовый отжиг остаётся перспективным направлением для экспериментов. Существующие машины для квантового отжига, такие как устройства компании D-Wave (см., например, [19]), уже подходят для исследований и могут быть использованы для изучения возможностей реализации машин с передачей данных из будущего.

5.4. Итог

Реализация машины Тьюринга с каналом передачи данных во времени возможна с использованием различных подходов, включая детерминированный поиск с возвратом, методы отжига и, в частности, квантовый отжиг. Каждый из этих методов имеет свои преимущества и ограничения, и выбор конкретного подхода зависит от характеристик решаемой задачи и доступных вычислительных ресурсов. Квантовый отжиг, в частности, остаётся перспективным направлением, хотя его практическая эффективность требует дальнейшего изучения.

6. Конструктивные распознаватели: пример решения задачи на машине с передачей данных во времени

6.1. Неконструктивность классических распознавателей

Классические алгоритмы распознавания обычно неконструктивны: они выдают только бинарный ответ (ДА или НЕТ) без объяснения результата. Например, на вопрос от том, существует ли y , удовлетворяющее условию $P(x, y)$, такие алгоритмы лишь подтверждают или опровергают существование такого y , не предоставляя сам объект y при положительном ответе или доказательство его отсутствия при отрицательном. Это ограничивает их использование в задачах, где требуется обоснование решения.

6.2. Конструктивные задачи в классе $NP \cap co-NP$

Машина Тьюринга с каналом передачи данных во времени позволяет реализовать конструктивные распознаватели, выдающие не только ответ, но и сертификат — доказательство или свидетельство результата. Мы рассматриваем симметричные задачи класса $NP \cap co-NP$, где свойство $Q(x)$ истинно, если существует сертификат y такой, что $Q_1(x, y)$, и ложно, если существует сертификат z такой, что $Q_0(x, z)$, причём размеры y и z полиномиально ограничены относительно $|x|$.

6.3. Пример: проверка наличия нетривиального делителя

Рассмотрим задачу определения, имеет ли число x нетривиальный делитель y такой, что $1 < y \leq d$ (порог d задан). Это свойство $Q(d, x)$:

- Если $Q(d, x) = 1$, то сертификат — делитель y , проверяемый делением за полиномиальное время.
- Если $Q(d, x) = 0$, то сертификат — факторизация x на простые множители, все $> d$, проверяемая за полиномиальное время (например, с использованием теста Миллера [20] при допущении обобщённой гипотезы Римана).

6.4. Реализация на машине с передачей данных во времени

Конструктивный распознаватель на машине с передачей данных во времени работает следующим образом:

(1) *Получение данных из будущего:*

- На первом шаге машина запрашивает с меткой 0 пару (b, C) , где $b \in \{0, 1\}$ — ответ, а C — сертификат (делитель или факторизация).
- Ответ b немедленно записывается на выходную ленту (если не требуется выдача ответа с фиксированной задержкой, то можно выдать и сертификат, тогда распознаватель получится конструктивным).

(2) *Проверка сертификата:*

- Если $b = 1$, то проверяется, что $C = y$ делит x и $1 < y \leq d$.
- Если $b = 0$, то проверяется, что C — факторизация x на простые множители $> d$, и их произведение равно x .

(3) *Самосогласование:*

- После проверки, если сертификат корректен, то (b, C) записывается на коммуникационную ленту с меткой 0.
- При некорректности создаётся противоречие, и ветвь вычислений отбрасывается.

Нетрудно видеть, что эта машина работает полиномиальное время.

Примеры выполнения:

Вход: $x = 15, d = 4$.

Шаг 1: Импорт $(1, 3)$ из будущего, вывод 1.

Шаг 2: Проверка: $15 \div 3 = 5, 1 < 3 \leq 4$, корректно.

Шаг 3: Передача $(1, 3)$ с меткой 0, самосогласование выполнено.

Вход: $x = 17, d = 4$.

Шаг 1: Импорт $(0, \{17\})$, вывод 0.

Шаг 2: Проверка: 17 — простое (тест Миллера), $17 > 4$, корректно.

Шаг 3: Передача $(0, \{17\})$, самосогласование выполнено.

6.5. Особенности и преимущества

Линейность: Процесс детерминирован.

Мгновенный отклик: Ответ доступен на первых шагах благодаря передаче из будущего.

Прозрачность: Если выдаётся сертификат, то это делает результат интерпретируемым, усиливая наглядность вычислений.

На рисунке 3 представлена схема работы конструктивного распознавателя: данные (b, C) передаются из конца вычислений в начало для немедленного вывода и последующей проверки. Блок-схема процесса от импорта результата до проверки и самосогласованной передачи данных дана на рисунке 4.



Рисунок 3. Схема работы конструктивного распознавателя

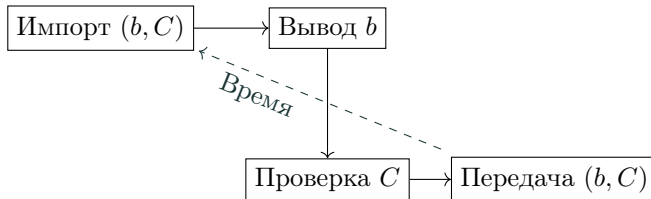


Рисунок 4. Блок-схема процесса

7. Действия для передачи данных во времени

Для преобразования обычного языка программирования в язык, поддерживающий передачу данных во времени, необходимо добавить две новые операции: передача данных в прошлое и приём данных из будущего. Эти действия позволяют машине взаимодействовать как с будущими, так и с прошлыми состояниями вычислительного процесса, предоставляя новые возможности для программирования. В данном разделе мы рассмотрим эти действия и их роль в организации вычислений с передачей данных во времени.

7.1. Действие передачи данных в прошлое

Действие передачи данных в прошлое позволяет машине записать данные в определённый момент времени, которые могут быть использованы в прошлых вычислениях. Это действие формализуется следующим образом:

Формальное описание:

- $transfer(l, d)$: передаёт объект d в момент времени l .
 - l — метка момента времени, для который передаются данные.
 - d — передаваемый объект (например, бит, число или строка).

Принцип работы:

- (1) Машина записывает объект d в канал передачи данных во времени с меткой времени l .
- (2) Данные становятся доступными для использования в прошлых шагах вычислений, связанных с моментом времени l .
- (3) Если запись с меткой l уже существует, возникает ошибка.

Пример: Если машина выполняет действие $transfer(1, 1)$, значение 1 передаётся с меткой 1. В прошлых вычислениях, связанных с 1, это значение может быть использовано для принятия решений.

7.2. Действие приёма данных из будущего

Действие приёма данных из будущего позволяет машине получить данные из определённого момента в будущем, что делает возможным использование будущей информации в текущих вычислениях. Это действие формализуется следующим образом:

Формальное описание:

- $accept(l)$: принимает данные из момента времени с меткой l .
 - l — имя момента времени, из которого запрашиваются данные.

Принцип работы:

- (1) Машина запрашивает данные из канала передачи данных во времени с меткой времени l .
- (2) Если данные существуют, они возвращаются и могут быть использованы в текущих вычислениях.
- (3) Если данные отсутствуют, возникает ошибка.

Пример: Если машина выполняет действие $accept(1)$, она получает значение d , которое было передано с меткой 1 с помощью действия $transfer(1, d)$. Это значение затем может быть использовано для принятия решений в текущих вычислениях.

7.3. Взаимодействие передач данных

Действия передачи и приёма данных работают совместно, обеспечивая связь между текущими и будущими состояниями вычислительного процесса. Это взаимодействие позволяет машине получать данные из будущего для принятия решений в текущих вычислениях. Важно отметить, что объём данных, передаваемых во времени, является важным показателем сложности вычисления. Например, при реализации процесса поиском с возвратом время работы замедляется экспоненциально с двоичным показателем объёма данных в битах, так как для каждого бита данных фактически приходится ввести две ветви вычислений. Подобные сложности возникают и в методах отжига.

7.4. Итог

Действия передачи и приёма данных во времени являются ключевыми элементами нашей вычислительной модели. В принципе, мы можем включить эти возможности в любой язык программирования, превратив его в язык управления машиной с передачей данных из будущего в прошлое. Эти инструменты предоставляют новые возможности для создания интуитивных вычислительных систем.

8. Распараллеливание вычислительного процесса

В данном разделе описывается параллельная машина с передачей данных во времени — расширение нашей машины с дополнительной возможностью, направленной на увеличение её вычислительной мощности: выполнение другого экземпляра машины с определёнными входными данными. Строка входных данных для нового процесса помещается на специальную ленту аргументов. Параллельный процесс запускается при переходе машины в специальное состояние запуска параллельного процесса. Новый процесс начинается с особого начального состояния подпроцесса. Изначально новый процесс выполняется до тех пор, пока не достигнет состояния, в котором выводит результат (в то время как старый процесс ожидает). Этот результат затем передаётся с выходной ленты нового процесса на ленту аргументов старого процесса. Затем оба процесса продолжают выполняться параллельно, при этом старый процесс больше не взаимодействует с новым процессом. Параллельное выполнение нового процесса необходимо для передачи данных в прошлое внутри него, что обеспечивает корректность результата, производимого этим процессом.

Новый процесс может быть запущен столько раз, сколько необходимо, основным процессом, либо порождёнными подпроцессами. Поскольку подпроцессы могут запускать свои собственные подпроцессы параллельно и независимо, количество подпроцессов может расти экспоненциально по мере работы машины в параллельном режиме.

Общее параллельное время работы машины считается временем до завершения всех параллельных процессов, как если бы они выполнялись синхронно, шаг за шагом.

ТЕОРЕМА 3. На параллельной машине с передачей данных во времени можно за полиномиальное время промоделировать работу обычной машины Тьюринга с полиномиальной памятью (что соответствует классу сложности PSPACE).

ДОКАЗАТЕЛЬСТВО. Рассмотрим машину Тьюринга с известной оценкой её емкостной сложности в классе PSPACE. Выполним следующее.

- (1) Сначала оценим время работы этой машины (экспоненциальное от объёма памяти).
- (2) Запустим подпроцесс для моделирования работы машины в течение половины этого времени (время моделирования передаётся подпроцессу вместе с начальной конфигурацией процесса).
- (3) Получив результат и позволив подпроцессу продолжить для его формирования в будущем, немедленно запустим новый подпроцесс для моделирования работы исходной машины в течение оставшейся второй половины выделенного времени, передав этому подпроцессу конфигурацию, полученную от первого подпроцесса.
- (4) Выведем результат.
- (5) Подпроцессы продолжают работу аналогичным образом, пока выделенное время не сойдётся к заранее определённой константе.

Нетрудно видеть, что глубина вызовов подпроцессов здесь полиномиальна, и общее параллельное время работы также полиномиально. $\square$

ТЕОРЕМА 4. Работа параллельной машины (с передачей данных во времени) за полиномиальное время может быть промоделирована обычной машиной Тьюринга с полиномиальным объёмом памяти.

ДОКАЗАТЕЛЬСТВО. Обратное утверждение следует из того, что работу такой параллельной машины можно промоделировать на обычной машине Тьюринга с использованием полиномиального объёма памяти. Это

достигается за счёт последовательного выполнения всех запущенных процессов, так как они не взаимодействуют друг с другом. Поскольку общее параллельное время моделируемой машины ограничено полиномиально, глубина вызовов подпроцессов также ограничена полиномиально. Таким образом, общий объём памяти для моделирования также остаётся полиномиальным. $\square$

9. Прямой доступ к входным данным

В данном разделе мы модифицируем модель нашей машины, добавив возможность работы с большими данными через прямой доступ к ячейкам входной ленты. Под большими данными понимаются данные, которые невозможно последовательно просмотреть в рамках текущих временных ограничений. Механизм прямого доступа позволяет быстро обращаться к удалённым ячейкам, что позволяет преодолеть указанное ограничение. Это достигается следующим образом: вводится специальная адресная лента, на которой записывается двоичное представление номера ячейки на входной ленте. Головка входной ленты мгновенно перемещается к указанной ячейке. Другие механизмы перемещения этой головки не предусмотрены. Мы предполагаем, что все запущенные подпроцессы работают с одними и теми же входными данными. Таким образом, мы определили параллельную машину с передачей данных во времени и с прямым доступом ко входным данным.

Если заменить PSPACE на LOGSPACE в предыдущем разделе, мы получим следующую теорему.

ТЕОРЕМА 5. На параллельной машине с передачей данных во времени и с прямым доступом к входным данным работа обычной машины Тьюринга с логарифмическим объёмом памяти (LOGSPACE) может быть промоделирована за логарифмическое время (LOGTIME).

ДОКАЗАТЕЛЬСТВО. Рассуждение аналогично доказательству Теоремы 3, но с заменой полиномиальных ограничений на логарифмические. Поскольку доступ к входным данным происходит за постоянное время, общее время выполнения остаётся логарифмическим. $\square$

Обратное утверждение проблематично, поскольку прямое моделирование логарифмического времени на нашей машине с использованием обычной машины Тьюринга требует объёма памяти, полиномиального от логарифма размера исходных данных. Однако аналогичный подход может быть использован для доказательства следующей теоремы.

ТЕОРЕМА 6. *Класс $POLYLOGTIME$ на параллельной машине с передачей данных во времени и с прямым доступом к входным данным совпадает с классом $POLYLOGSPACE$ для обычных машин Тьюринга.*

ДОКАЗАТЕЛЬСТВО. Утверждение основано на том, что работа нашей машины с прямым доступом может быть промоделирована на обычной машине Тьюринга с использованием полилогарифмической памяти. Таким образом, общее время моделирования остаётся полилогарифмическим. $\square$

Заключение

В данной работе предложена принципиально новая вычислительная модель, расширяющая классические представления о вычислительных процессах за счёт механизма передачи данных состояниями машины в различные моменты времени. Основные результаты исследования заключаются в следующем:

- (1) *Классы сложности и временные ограничения:* Доказано, что машина с передачей данных из будущего, без ограничений по времени, распознаёт в точности разрешимые множества с постоянной задержкой ответа. При полиномиальных ограничениях по времени модель соответствует классу задач из пересечения NP и $co-NP$. Эти результаты показывают, как контролируемые нарушения причинного порядка могут помогать вычислениям.
- (2) *Распараллеливание и емкостная сложность:* Расширение модели возможностью запуска подпроцессов, наследующих входные данные, позволяет охватить класс $PSPACE$ за полиномиальное время, что демонстрирует, как временная сложность преобразуется в емкостную.
- (3) *Прямой доступ и эффективность обработки данных:* Модификация с адресуемым доступом ко входной ленте и логарифмической временной сложностью обеспечивает соответствие классу $LOGSPACE$, что особенно актуально для задач обработки больших данных. Этот результат поддерживается механизмом мгновенного перемещения головки входной ленты, устраняющим линейные задержки.
- (4) *Практическая реализация:* Предложены два подхода к практической реализации модели:
 - *Детерминированный поиск с возвратом* — систематический поиск самосогласованных вычислительных траекторий;

- *Методы отжига* (включая квантовые системы) — поиск состояний с минимальной энергией противоречий.

Эксперименты с квантовыми отжигателями, такими как D-Wave, могут стать следующим шагом для оценки физической реализуемости модели.

Перспективы дальнейших исследований охватывают как теоретические, так и практические аспекты:

- Оптимизация объёма передаваемых данных для снижения вычислительной нагрузки;
- Разработка специализированных языков программирования с операторами для передачи данных во времени;
- Анализ связи с квантовыми вычислениями.

Данная работа актуальна не только для теоретической информатики, но и для междисциплинарных исследований. Она демонстрирует, что связь состояний во времени может стать ключом к решению задач, недоступных для традиционных систем. Предложенная модель может служить мостом между теорией сложности и физическими реализациями, открывая новые возможности для алгоритмов.

Список использованных источников

- [1] Cook S. A. *The complexity of theorem-proving procedures* // *Logic, automata, and computational complexity: The works of Stephen A. Cook*, ed. Kapron B.C., New York: ACM.— 2023.— ISBN 979-8-4007-0779-7.— Pp. 143–152.  [↑30](#)
- [2] Karp R. M. *Reducibility among combinatorial problems* // *50 Years of Integer Programming 1958–2008. From the Early Years to the State-of-the-Art*, eds. Jünger M., et al., Berlin–Heidelberg: Springer.— 2010.— ISBN 978-3-540-68274-5.— Pp. 219–241.  [↑30](#)
- [3] Garey M. R., Johnson D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Series of Books in the Mathematical Sciences.— San Francisco: Freeman.— 1979.— ISBN 978-0716710455.— 340 pp. [↑31](#)
- [4] Aaronson S. *Guest column: NP-complete problems and physical reality* // *ACM Sigact News*.— 2005.— Vol. **36**.— No. 1.— Pp. 30–52.  [↑31](#)
- [5] Deutsch D. *Quantum theory, the Church–Turing principle and the universal quantum computer* // *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*.— 1985.— Vol. **400**.— No. 1818.— Pp. 97–117.  [↑31](#)

- [6] Brun T. A. *Computers with closed timelike curves can solve hard problems efficiently* // Foundations of Physics Letters.– 2003.– Vol. **16**.– No. 3.– Pp. 245–253.  ^{↑31}
- [7] Aaronson S., Watrous J. *Closed timelike curves make quantum and classical computing equivalent* // Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences.– 2009.– Vol. **465**.– No. 2102.– Pp. 631–647.  ^{↑31}
- [8] Papadimitriou Ch. *Computational Complexity*.– Reading, Massachusetts: Addison-Wesley.– 1994.– ISBN 978-0201530827.– 523 pp. ^{↑31}
- [9] Arora S., Barak B. *Computational Complexity: A Modern Approach*.– New York: Cambridge University Press.– 2009.– ISBN 978-0521424264.– 594 pp. ^{↑31}
- [10] Nielsen M. A., Chuang I. L. *Quantum computation and quantum information*.– New York: Cambridge University Press.– 2010.– ISBN 9781107002173.– 702 pp.  ^{↑31}
- [11] Grover L. K. *A fast quantum mechanical algorithm for database search* // STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing (22–24 May 1996, Pennsylvania, USA), New York: ACM.– 1996.– ISBN 978-0-89791-785-8.– Pp. 212–219.  ^{↑31}
- [12] Shor P. W. *Algorithms for quantum computation: discrete logarithms and factoring* // Proceedings 35th Annual Symposium on Foundations of Computer Science (20–22 November 1994, Santa Fe, NM, USA).– IEEE.– 1994.– ISBN 0-8186-6580-7.– Pp. 124–134.  ^{↑31}
- [13] Harrow A. W., Montanaro A. *Quantum computational supremacy* // Nature.– 2017.– Vol. **549**.– No. 7671.– Pp. 203–209.  ^{↑31}
- [14] Iqbal K., Ahmad O., Vandika A. Y. *Quantum computing and its implications for complex system analysis* // Research of Scientia Naturalis.– 2024.– Vol. **1**.– No. 5.– Pp. 238–247.  ^{↑31}
- [15] Orlicki J. I. *Time-travelling Turing Machines and the self-consistent Halting Problem*.– 2024.  ^{↑31}
- [16] Gödel K. *An example of a new type of cosmological solutions of Einstein's field equations of gravitation* // Rev. Mod. Phys.– 1949.– Vol. **21**.– No. 3.– Pp. 447–450.  ^{↑32}
- [17] Bacon D. *Quantum computational complexity in the presence of closed timelike curves* // Phys. Rev. A.– 2004.– Vol. **70**.– No. 3.– id. 032309.  ^{↑32}
- [18] Wolpert D. H. *Physical limits of inference* // Physica D: Nonlinear Phenomena.– 2008.– Vol. **237**.– No. 9.– Pp. 1257–1281.  ^{↑32}
- [19] King A. D., Raymond J., Lanting T., Harris R., Zucca A., Altomare F., Berkley A. J., Boothby K., Ejtemaee S., Enderud C., Hoskinson E., Huang Sh., Ladizinsky E., MacDonald A. J. R., Marsden G., Molavi R., Oh T., Poulin-Lamarre G., Reis M., Rich Ch., Sato Y., Tsai N., Volkmann M., Whittaker J. D., Yao J., Sandvik A. W., Amin M. H. *Quantum critical dynamics in a 5,000-qubit programmable spin glass* // Nature.– 2023.– Vol. **617**.– No. 7959.– Pp. 61–66.  ^{↑32, 42}

- [20] Miller G. L. *Riemann's hypothesis and tests for primality* // *STOC '75: Proceedings of the seventh annual ACM symposium on Theory of computing* (5–7 May 1975, Albuquerque, New Mexico, USA), New York: ACM.— ISBN 978-1-4503-7419-4.— С. 234–239. doi ↑43

Поступила в редакцию 15.03.2025;
одобрена после рецензирования 02.04.2025;
принята к публикации 02.04.2025;
опубликована онлайн 21.04.2025.

Рекомендовал к публикации

д.ф.-м.н. Н. Н. Непейвода

Информация об авторах:



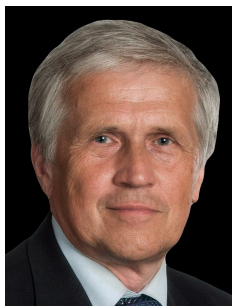
Милад Джудакизаде

Аспирант в области искусственного интеллекта и машинного обучения, исследования сосредоточены на вычислительной сложности, машинном обучении и логике в компьютерных науках.



0000-0002-6167-6237

e-mail: joudakizadeh@mail.ru



Анатолий Петрович Бельтюков

Доктор физико-математических наук, профессор. Научные интересы включают вычислительную сложность, классы сложности, субрекурсивные иерархии, интуиционистскую математику, конструктивные системы, механизацию доказательств, сложность доказательств, модели вычислений, субструктурные логики, слабые арифметики и логику в компьютерных науках.



0000-0002-3433-9067

e-mail: belt.udsu@mail.ru

Авторы внесли равный вклад в подготовку публикации.

Декларация об отсутствии личной заинтересованности: благополучие авторов не зависит от результатов исследования.