Министерство науки и высшего образования
Российской Федерации

Тверской государственный университет

# Всероссийская научная конференция

# «Математические основы информатики и информационно-коммуникационных систем»

## Сборник трудов

Тверь
3–8 декабря 2021 г.

Под редакцией С. М. Дудакова и Б. Н. Карлова

Тверь 2021

**УДК 004, 510, 519**
**ББК 22.1, 32.97я43**
**В85**

Тверской государственный университет, г. Тверь

Математический институт им. В. А. Стеклова
Российской академии наук, г. Москва

Математический центр мирового уровня
«Математический институт им. В. А. Стеклова
Российской академии наук» (МЦМУ МИАН), г. Москва

**В85** Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. Тверь, 3–8 декабря 2021 г. / Под ред. С. М. Дудакова и Б. Н. Карлова. — Тверь : ТвГУ, 2021. — 290 с.

ISBN 978-5-7609-1682-2

Сборник содержит материалы, представленные на всероссийской научной конференции «Математические основы информатики и информационно-коммуникационных систем».

УДК 004, 510, 519
ББК 22.1, 32.97я43

**ISBN 978-5-7609-1682-2**

# Содержание

# Всероссийская научная конференция

# «Математические основы информатики и информационно-коммуникационных систем»

## Тверь, 3–8 декабря

Тверской государственный университет

МИАН

SIMC

Steklov International Mathematical Center

SIMONS FOUNDATION

Ростелеком Информационные технологии

## Организаторы

Тверской государственный университет, г. Тверь

Математический институт им. В. А. Стеклова
Российской академии наук, г. Москва

Математический центр мирового уровня
«Математический институт им. В. А. Стеклова
Российской академии наук» (МЦМУ МИАН), г. Москва

## Финансовая поддержка

Конференция проводится при финансовой поддержке

Фонда Саймонса и Минобрнауки России
(грант на создание и развитие МЦМУ МИАН,
соглашение №075-15-2019-1614);

ООО «Ростелеком Информационные Технологии»

# Программный комитет конференции «Математические основы информатики и информационно-коммуникационных систем»

**Председатель программного комитета:**

- Дудаков Сергей Михайлович, декан факультета прикладной математики и кибернетики.

**Члены программного комитета:**

- Беклемишев Лев Дмитриевич, академик РАН, заведующий отделом Математического института им. В. А. Стеклова РАН, Москва;
- Максимова Лариса Львовна, главный научный сотрудник Института математики им. Соболева СО РАН, Новосибирск;
- Одинцов Сергей Павлович, ведущий научный сотрудник Института математики им. Соболева СО РАН, Новосибирск;
- Хохлов Юрий Степанович, профессор Московского государственного университета им. М. В. Ломоносова, Москва;
- Сухомлин Владимир Александрович, профессор Московского государственного университета им. М. В. Ломоносова, Москва;
- Пентус Мати Рейнович, профессор Московского государственного университета им. М. В. Ломоносова, Москва;
- Ломазова Ирина Александровна, профессор Национального научно-исследовательского университета «Высшая школа экономики», Москва;

- Язенин Александр Васильевич, заведующий кафедрой информационных технологий Тверского государственного университета, Тверь;
- Соколов Валерий Анатольевич, заведующий кафедрой теоретической информатики Ярославского государственного университета им. Демидова, Ярославль;
- Башкин Владимир Анатольевич, профессор Ярославского государственного университета им. Демидова, Ярославль;
- Кузьмин Егор Владимирович, профессор Ярославского государственного университета им. Демидова, Ярославль;
- Артёмов Сергей Николаевич, профессор университета City University of New York, США, Нью-Йорк;
- Дехтярь Александр Михайлович, профессор университета California Polytechnic State University, США, Сан-Луис-Обиспо;
- Зильбер Борис Иосифович, профессор университета University of Oxford, Mathematical Institute, Великобритания, Оксфорд.

# Организационный комитет конференции «Математические основы информатики и информационно-коммуникационных систем»

**Председатель организационного комитета:**

- Дудаков Сергей Михайлович, декан факультета прикладной математики и кибернетики.

**Секретарь:**

- Карлов Борис Николаевич, доцент кафедры информатики.

**Члены организационного комитета:**

- Язенин Александр Васильевич, заведующий кафедрой информационных технологий;
- Солдатенко Илья Сергеевич, доцент кафедры информационных технологий;
- Захарова Ирина Владимировна, доцент кафедры математической статистики и системного анализа;
- Рыбаков Михаил Николаевич, доцент кафедры функционального анализа и дифференциальной геометрии;
- Секорин Всеслав Станиславович, аспирант кафедры информатики;
- Михайлова Марина Владимировна, ведущий документовед деканата факультета прикладной математики и кибернетики;
- Лордкипанидзе Ольга Юрьевна, старший диспетчер факультета прикладной математики и кибернетики.

УДК 929
AMS MSC2020: 01A70, 01A72

# О научной школе по математическим основам информатики в Тверском государственном университете[1]

**Дудаков С. М., Карлов Б. Н., Дадеркин Д. О.**

Тверской государственный университет

Аннотация. Мы кратко описываем жизненный путь и основные достижения основателей научной школы по математическим основам информатики в Тверском государственном университете: Михаила Абрамовича Тайцлина и Михаила Иосифовича Дехтяря — памяти которых посвящена конференция.

Ключевые слова: математические основы информатики, Тверской государственный университет, Михаил Абрамович Тайцлин, Михаил Иосифович Дехтярь.

## Введение

Факультет прикладной математики и кибернетики был образован в Тверском государственном университете в 1977 году. Было это связано со все более возрастающей ролью приложений математики к различным задачам и использованием вычислительной техники для их решения. Такая тенденция в целом характерна для того времени.

Но далеко не сразу на факультете сформировалась устойчивая традиция и научная школа в сфере математических оснований информатики. Первоначально дисциплины, связанные с информатикой и вычислительной техникой, преподавались, в основном, выходцами из разного рода военных учреждений. Это не могло привести к появлению научно-педагогического коллектива, который занимался как преподаванием, так и научными исследованиями, соответствующими мировому уровню.

[1]Работа первых двух авторов поддержана РФФИ, проект 20-01-00435.

Ситуация изменилась в середине 80-х годов XX в. когда на факультет пришли два выдающихся выпускника Новосибирского государственного университета: Михаил Абрамович Тайцлин и Михаил Иосифович Дехтярь. Именно с их именами и связано становление научных и педагогических традиций Тверского государственного университета в области теоретических основ информатики.

## 1. Михаил Абрамович Тайцлин (1936–2013)

Михаил Абрамович Тайцлин, [5] — ученик знаменитого академика А. И. Мальцева (1909–1967). Он продолжал исследования своего учителя по алгебре и логике, например, ему принадлежит ряд результатов в теории полугрупп, колец и их элементарных теорий. Однако постепенно область его интересов расширяется за счёт интенсивно развивающейся вычислительной техники и математических оснований информатики. Когда он в 1976 году переходит в Казахский университет, он там уже сразу возглавляет кафедру математического обеспечения ЭВМ. В этот период выходят его работы посвященные динамическим логикам и другим разделам математики, которые непосредственно связаны с теорией программирования.

В 1984 году М. А. Тайцлин приходит в Калининский (сейчас — Тверской) государственный университет, где становится заведующим кафедрой алгоритмических языков и системного программирования (позже переименованной в кафедру информатики) и возглавляет ее до 2009 года, когда ему пришлось оставить этот пост из-за проблем со здоровьем. Однако он до конца своей жизни оставался профессором кафедры информатики, активно участвуя в образовательной и научной жизни факультета.

Влияние, которое оказал М. А. Тайцлин в ТвГУ на развитие преподавания и научных исследований не только в сфере теоретических основ информатики, но и всех связанных с этим областей, трудно переоценить. За время работы он исследовал самые разные задачи: динамические логики, сложность вычислений, выразительную силу логических языков, теорию конечных моделей, логическое программирование и многие другие. И в каждой им и его учениками были получены замечательные результаты.

Мы не будем перечислять все достижения, которые получил Михаил Абрамович, за этим можно обратиться к [5], а только укажем

те, с которыми авторам лично пришлось иметь дело. Так, им был получен ряд результатов связанных с коммутативными полугруппами. В частности, показано, что теория всякой конечно определенной коммутативной полугруппы разрешима [8]. Следующей областью исследований, которую мы хотим упомянуть, является линейная логика Ж.-И. Жирара. М. А. Тайцлиным, М. И. Дехтярем (о нем — ниже) и их учениками были получены результаты [10], которые связаны с использованием мультипликативных фрагментов линейной логики для моделирования параллельных процессов и сложности связанных с этим задач. В этих же работах предложена семантика мультипликативного фрагмента, как задачи о «бартерных сделках».

Следующая группа результатов М. А. Тайцлина связана с применением логических языков в базах данных и системах искусственного интеллекта. Одной из таких задач было исследование синтаксиса логических программ (даталога) на предмет его «безопасности», то есть гарантированной возможности получения результатов этих программ за конечное время. То, что в общем виде эта задача не имеет решения, было известно. Однако оставалась вероятность, что проблема может быть положительно решена хотя бы для некоторой части, так называемых, стратифицированных, программ. Но М. А. Тайцлиным совместно с Д. А. Архангельским было показано, что и это невозможно [9]. Еще одно направление исследований М. А. Тайцлина — динамические логики [6]. В частности, им совместно с А. П. Столбоушкиным, а позже — с Д. О. Дадеркиным, получены результаты о значении недетерминизма и рекурсии.

Наконец, скажем о работах М. А. Тайцлина и его учеников в области, которая имеет прямое отношение к теории языков запросов баз данных. Проблема, решению которой посвящены эти исследования, заключается в следующем: какое влияние оказывают на выразительные возможности языка запросов отношения и функции «внешние» по отношению к базе данных, то есть определенные не для ее элементов, а в целом для универсума, из которого эти элементы берутся? Как было показано Ю. Ш. Гуревичем, наличие линейного порядка на универсуме действительно расширяет эти возможности. Однако дальнейшее обогащение сигнатуры или не приводило к увеличению выразительной силы, или же сразу приводило к неразрешимой теории. В результате исследований М. А. Тайцлина, О. В. Белеградека, А. П. Столбоушкина и позже С. М. Дудакова были найдены доста-

точные условия, позволяющие утверждать невозможность этого увеличения во многих практически важных случаях, например, для классических числовых систем (см. обзор [3]).

Кроме научной работы, мы хотим отметить преподавательский и методический талант М. А. Тайцлина. Сразу после прихода в ТвГУ им были модифицированы курсы, связанные с обучением информатике и программированию, в строгом и последовательном стиле. Фактически эти дисциплины читаются в ТвГУ уже более 30 лет, следуя в общих чертах той программе, которая была разработана М. А. Тайцлиным. Им же были разработаны и программы основных курсов по дискретной математике и по математическим основам информатики. Последняя программа легла в основу курсов по математической логике и теории алгоритмов, а также — по теории автоматов и формальных языков. Они тоже используются до настоящего времени с минимальными коррективами. Упомянем и разработанные М. А. Тайцлиным совместно с коллегами учебники по математической логике [4] и математическим основам информатики [7], которые затем послужили прообразом для других книг.

## 2. Михаил Иосифович Дехтярь (1946–2018)

Другой крупной фигурой, определившей развитие школы по математическим основам информатики в ТвГУ, был Михаил Иосифович Дехтярь (1946–2018), [16]. Он являлся учеником другого представителя новосибирской школы — Б. А. Трахтенброта (1921–2016). М. И. Дехтярь оказался в Калинине (Твери) даже раньше М. А. Тайцлина — в 1981 году, так как вынужден был покинуть новосибирский академгородок после эммиграции Б. А. Трахтенброта. Но в Калинине он первоначально оказался не по академической линии, а в качестве специалиста в области разработки программных продуктов (СПКБ СУ, Центрпрограммсистем). К работе в ТвГУ его привлек М. А. Тайцлин в 1987 году, и с тех пор М. И. Дехтярь оставался сотрудником кафедры алгоритмических языков и системного программирования (затем — информатики). Только в 2015 году он был вынужден оставить работу в ТвГУ из-за сильно ухудшившегося состояния здоровья.

Работы М. И. Дехтяря тоже охватывают широкий круг различных тем. Первые его исследования были посвящены проблемам сложно-

сти: колмогоровской и вычислительной, а также — их взаимосвязи. В частности, один из результатов показывает, что вычислительно сложные задачи не могут быть «аппроксимируемы» с низкой колмогоровской сложностью [12]. Другой класс работ М. И. Дехтяря посвящен исследованию алгоритмов для работы со строками, в частности, языку РЕФАЛ, разработанному именно для обработки текстовой информации. Им была разработана концепция интерпретации некоторых конструкций РЕФАЛа и доказана NP-полнота соответствующих задач [2]. Совместная с А. М. Дехтярем работа [11] посвящена поиску корректной семантики вероятностных логических программ.

Следующая группа задач была посвящена исследованию интеллектуального поведения различных систем. В частности, одной из таких проблем является «разумное» обновление базы данных, которое предполагает не только выполнение действия явно заданных пользователем, но и дополнительные операции, которые должны быть выведены из ограничений целостности. М. И. Дехтярем совместно с А. Я. Диковским, Н. Спиратосом и С. М. Дудаковым были получены оценки сложности для решения указанной задачи, а в случаях, когда эти оценки были приемлемыми, — разработаны соответствующие алгоритмы для их решения. Другой пример системы — это множество агентов, каждый из которых действует независимо от других, хотя они могут обмениваться сообщениями. Появляется вопрос о траектории поведения такой мультиагентной системы и возможности или вероятности достижения ею каких-то целей. Совместные работы М. И. Дехтяря, А. Я. Диковского и М. К. Валиева позволили дать ответ на ряд вопросов в этой сфере [15].

Еще одной темой для исследований стали способы определения и порождения языков. Классические грамматики, принадлежащие иерархии Н. Хомского, не всегда подходят, особенно для естественных языков. Они оказываются то недостаточно выразительными, то слишком мощными (и неразрешимыми). Поэтому появился вопрос о поиске новых вариантов формализации языков. Одной из таких формализаций стали предложенные в работах М. И. Дехтяря совместно с А. Я. Диковским и Б. Н. Карловым категориальные грамматики зависимостей. Они способны выражать понятия, которые часто встречаются в естественных языках, например, связи между «далекими» словами. Далее эта концепция была ещё расширена,

были введены мультимодальные категориальные грамматики зависимостей. Для указанных классов грамматик были предложены и распознающие их автоматы, а также — получены оценки сложности анализа языков с применением таких грамматик.

Михаил Иосифович, как и М. А. Тайцлин, оставил свой глубокий след не только в научных исследованиях, но и в учебно-методической сфере. В частности, им был разработан учебник по дискретной математике [1]. Также он сформировал структуру курса по построению эффективных алгоритмов и анализу их сложности, который с небольшими изменениями до настоящего времени следует этому стилю.

* * *

Будучи учениками Михаила Абрамовича Тайцлина и Михаила Иосифовича Дехтяря, авторы всегда будут искренне благодарны своим учителям за все те знания и опыт, которыми они делились со своими учениками.

**Список литературы**

[1] *Дехтярь, М. И.* Лекции по дискретной математике. — М. : Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007. — 259 с.

[2] *Дехтярь, М. И.* О семантике и доказательстве свойств Рефал-программ // Всесоюзная научная конференция «Проблемы совершенствования, тестирования, верификации и отладки программ». Рига'1986. — Рига, 1986. — С. 102–103.

[3] *Дудаков, С. М.* Трансляционные результаты для языков запросов в теории баз данных / С. М. Дудаков, М. А. Тайцлин // Успехи математических наук. — 2006. — Т. 61, № 2 (368). — С. 3–66.

[4] *Ершов, Ю. Л.* Математическая логика: лекции для студентов-математиков НГУ / Ю. Л. Ершов, Е. А. Палютин, М. А. Тайцлин — Новосибирск: НГУ, 1973. — 160 с.

[5] Михаил Абрамович Тайцлин (1936–2013) / Д. А. Архангельский, Б. С. Байжанов, О. В. Белеградек [et al.] // Сибирские электронные математические известия. — 2013. — Т. 10. — С. A54–A65.

[6] *Столбоушкин, А. П.* Динамические логики / А. П. Столбоушкин, М. А. Тайцлин // Кибернетика и вычислительная техника. — М.: Наука, 1986. — Вып. 2. — С. 180–230.

[7] *Столбоушкин, А. П.* Математические основания информатики / А. П. Столбоушкин, М. А. Тайцлин. — Тверь : Тверской государственный университет, 1998.

[8] *Тайцлин, М. А.* Об элементарных теориях коммутативных полугрупп // Алгебра и логика. — 1966. — Т. 5, № 4. — С. 55–89.

[9] *Arhangelsky, D. A.* A logic for data description / D. A. Arhangelsky, M. A. Taitslin // Logical Foundation of Computer Science, Pereslavl-Zalessky'89, LNCS'363. — Berlin : Springer-Verlag, 1989. — P. 2–11.

[10] *Archangelsky, D. A.* Linear logic for nets with bounded resources / D. A. Archangelsky, M. I. Dekhtyar, M. A. Taitslin // Annals of Pure and Applied Logic. — 1996. — № 78 (1–3). — P. 3–28.

[11] *Dekhtyar, A. M.* Possible Worlds Semantics for Probabilistic Logic Programs / A. Dekhtyar, M. I. Dekhtyar // Proc. of International Conference on Logic Programming (ICLP)' 2004, LNCS'3132. — Berlin : Springer-Verlag, 2004. — P. 137–148.

[12] *Dekhtjar, M. I.* Bounds on computational complexity and approximability of initial segments of recursive sets // Proc. 8th Symp. Mathematical Foundations of Computer Science, Olomouc (Czech.), LNCS'74. — Berlin : Springer-Verlag, 1979. — P. 277–283.

[13] *Dekhtyar, M. I.* Categorial dependency grammars / M. Dekhtyar, A. Dikovsky, B. Karlov // Theoretical Computer Science. — 2015. — № 579. — P. 33–63.

[14] *Dekhtyar, M. I.* Maximal State Independent Approximations to Minimal Real Change / M. Dekhtyar, A. Dikovsky, S. Dudakov, N. Spyratos // Annals of Mathematics and Artificial Intelligence. — 2001. — № 33 (2–4). — P. 157–204.

[15] *Dekhtyar, M. I.* Temporal verification of probabilistic multi-agent systems / M. I. Dekhtyar, A. J. Dikovsky, M. K. Valiev // Pillars of Computer Science, LNCS'4800 / Eds. Avron A., Dershowitz N., Rabinovich A. — Berlin : Springer-Verlag, 2008. — P. 256–265.

[16] *Dudakov, S. M.* Mikhail Iosifovich Dekhtyar (1946–2018) / S. M. Dudakov, B. N. Karlov // 9th Workshop PSSV : proceedings,

Yaroslavl, June, 21–22, 2018. — Yaroslavl : Yaroslavl State University, 2018. — P. 12–14.

### Библиографическая ссылка

*Дудаков, С. М.* О научной школе по математическим основам информатики в Тверском государственном университете / С. М. Дудаков, Б. Н. Карлов, Д. О. Дадеркин // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 12–19.
https://doi.org/10.26456/mfcsics-21-1

### Сведения об авторах

1. Сергей Михайлович Дудаков
Тверской государственный университет. Декан факультета прикладной математики и кибернетики

*Россия, 170002, Тверь, пер. Садовый, 35, ТвГУ*
*E-mail: sergeydudakov@yandex.ru*

2. Борис Николаевич Карлов
Тверской государственный университет. Доцент

*Россия, 170002, Тверь, пер. Садовый, 35, ТвГУ*
*E-mail: bnkarlov@gmail.com*

3. Дмитрий Ольгердович Дадеркин
Тверской государственный университет. Доцент

*Россия, 170002, Тверь, пер. Садовый, 35, ТвГУ*
*E-mail: d.daderkin@yandex.ru*

УДК 004.89
AMS MSC2020: 03B52

# Нечеткий когнитивный анализ и моделирование сложных систем и процессов[1]

**Борисов В. В.**

Филиал ФГБОУ ВО «Национальный исследовательский университет «МЭИ» в г. Смоленске

Аннотация. Охарактеризованы проблемы исследования сложных систем, процессов и проблемных ситуаций и представлены подходы к использованию нечетких когнитивных моделей для их анализа и моделирования, заключающиеся: во-первых, в предварительном анализе систем, процессов и проблемных ситуаций, результаты которого применяются для более углубленного исследования; во-вторых, в замене/модернизации отдельных компонентов моделей для расширения возможностей и улучшения свойств базовых моделей; в-третьих, в построении композиционных гибридных нечетких моделей, в которых отдельные нечеткие модели выполняют различные задачи по достижению общей цели. Приведены примеры реализации указанных подходов при исследовании систем, процессов и проблемных ситуаций.

Ключевые слова: Сложные системы и процессы, нечеткая когнитивная модель, композиционная гибридная модель.

Проблемы исследования сложных систем и процессов, определяющие целесообразность использования для их разрешения нечетких когнитивных моделей и методов, заключаются в следующем:

- разнородность объектов и компонентов, многообразие и разнотипность связей и взаимозависимостей между ними, не поддающиеся точному и детализированному описанию; сложность формализованного представления и анализа системы/процессов/ситуаций в рамках единой модели;

[1]Работа выполнена при финансовой поддержке Министерства науки и высшего образования РФ в рамках государственного задания № FSWF-2020-0019.

- неясность «границ» системы, проблемных ситуаций;
- сложность получения информации для построения «точных» моделей;
- доступность информации в виде экспертных данных, либо в эвристическом виде, либо ее недостаточность для задания аналитических зависимостей;
- оценка системных параметров с использованием различных шкал;
- сложность учета различных типов неопределенности.

В докладе рассмотрены следующие основные типы нечетких когнитивных моделей для исследования сложных систем и процессов:

- нечеткие когнитивные карты Б. Коско и их разновидности [10, 11];
- нечеткие когнитивные карты В. Силова [5];
- нечеткие продукционные когнитивные карты [8, 9];
- обобщенные продукционные нечеткие когнитивные модели [1];
- нечеткие реляционные когнитивные модели [1, 6];
- «совместимые» нечеткие когнитивные модели [7];
- нечеткие игровые и коалиционные когнитивные модели [2];
- нечетко-нейросетевые когнитивные темпоральные модели [3].

Особенностями рассмотренных нечетких когнитивных моделей, создающих основу для разрешения указанных выше проблем, являются:

- формализация и анализ проблемных ситуаций с обеспечением требуемого уровня достоверности анализа и моделирования;
- использование информации, измеряемой и оцениваемой с использованием различных шкал;

- учет непосредственного, опосредованного и агрегированного взаимовлияния системных и внешних факторов;
- нечеткая грануляция значений системных параметров, внешних факторов, целевых функций и ограничений;
- представление взаимозависимостей между объектами и компонентами в виде нечетких отношений взаимовлияния и единый подход к анализу с использованием методов нечеткой каузальной алгебры;
- наглядность и интерпретация процесса и результатов моделирования;
- учет различных типов неопределенности в рамках единой модели.

Вместе с тем, существующие методы и модели нечеткого когнитивного моделирования, как правило, ограничивается предварительным анализом систем, процессов и проблемных ситуаций, а именно: анализом отношений взаимовлияния системных и внешних факторов; оценкой системных характеристик; анализом устойчивости; прогнозной оценкой состояния; моделированием динамики [1].

В настоящее время предпринимаются усилия по приданию методам нечеткого когнитивного моделирования дополнительных возможностей за счет их интеграции с другими методами и моделями [4].

В докладе рассмотрены три подхода, ориентированные на расширение возможностей нечеткого когнитивного анализа и моделирование систем, процессов и проблемных ситуаций.

Первый подход заключается в использовании нечетких когнитивных моделей для предварительного анализа, результаты которого используются для последующего, более углубленного исследования систем, процессов и проблемных ситуаций.

Второй подход заключается в гибридизации моделей за счет замены или усовершенствования отдельных их компонентов компонентами других моделей для расширения возможностей и улучшения свойств.

Третий подход базируется на построении композиционных гибридных моделей, в которых отдельные нечеткие модели выполняют различные задачи по достижению общей цели. Приведены примеры

реализации указанных подходов при исследовании систем, процессов и проблемных ситуаций.

Рассмотрены примеры реализации описанных подходов для анализа и моделирования систем, процессов проблемных ситуаций.

**Список литературы**

[1] *Борисов, В. В.* Нечеткие модели и сети. 2-е изд. стереотип / В. В. Борисов, В. В. Круглов, А. С. Федулов. — М. : «Горячая линия — Телеком», 2018. — 284 с.

[2] *Борисов, В. В.* Анализ взаимодействий в сложных системах на основе нечетких когнитивных и игровых моделей / В. В. Борисов, Е. С. Устиненков // Нейрокомпьютеры: разработка и применение. — 2009. — № 8. — С. 4–12.

[3] *Борисов, В. В.* Метод многомерного анализа и прогнозирования состояния сложных систем и процессов на основе нечетких когнитивных темпоральных моделей / В. В. Борисов, В. С. Луферов. // Системы управления, связи и безопасности. — 2020. — № 2. — С. 1–23.

[4] *Борисов, В. В.* Систематизация нечетких и гибридных нечетких моделей // Мягкие измерения и вычисления. — 2020. — Т. 29, № 4. — С. 98–120.

[5] *Силов, В. Б.* Принятие стратегических решений в нечеткой обстановке. — М. : ИНПРО-РЕС, 1995. — 228 с.

[6] *Федулов, А. С.* Анализ нечетких реляционных когнитивных карт / А. С. Федулов, В. В. Борисов. // Нейрокомпьютеры: разработка, применение. — 2016. — № 7. — С. 7–14.

[7] *Borisov, V.* «Compatible» Fuzzy Cognitive Maps for Direct and Inverse Inference / V. Borisov, A. Fedulov, Ya. Fedulov // Proceedings of the 18th International Conference on Computer Systems and Technologies CompSysTech'17. — New York, N. Y. : ACM, 2017. — P. 20–27.

[8] *Carvalho, J. P.* Rule Based Fuzzy Cognitive Maps — Qualitative Systems Dynamics / J. P. Carvalho, J. A. B. Tomé // PeachFuzz 2000. 19th International Conference of the North American Fuzzy

Information Processing Society — NAFIPS (Cat. No.00TH8500). — Manhattan, N. Y. : IEEE, 2000. — P. 407–411.

[9] *Carvalho, J. P.* Rule Based Fuzzy Cognitive Maps in Socio-Economic Systems / J. P. Carvalho, J. A. B. Tomé // IFSA/EUSFLAT Conf. — 2009. — P. 1821–1826.

[10] *Kosko, B.* Fuzzy cognitive maps // International Journal of Man-Machine Studies. — 1986. — Vol. 24. — P. 65–75.

[11] *Thulukkanam, K.* Two New Fuzzy Models Using Fuzzy Cognitive Maps Model and Kosko Hamming Distance / K. Thulukkanam, R. Vasuki // Ultra Scientist. — 2015. — Vol. 27, № 1B. — P. 43–55.

**Библиографическая ссылка**

*Борисов, В. В.* Нечеткий когнитивный анализ и моделирование сложных систем и процессов // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 20–24.
https://doi.org/10.26456/mfcsics-21-2

**Сведения об авторах**

Вадим Владимирович Борисов
Филиал ФГБОУ ВО «Национальный исследовательский университет «МЭИ» в г. Смоленске. Профессор кафедры вычислительной техники, руководитель лаборатории интеллектуальных систем и высокопроизводительных вычислений

*214003, г. Смоленск, Энергетический проезд, д. 1*
*E-mail: vbor67@mail.ru*

УДК 510.53, 512.54.05
AMS MSC2020: 20M05, 03B30

# Алгоритмические проблемы для уравнений в свободных группах и полугруппах с ограничениями на решения

**Дурнев В. Г., Зеткина А. И.**

Ярославский государственный университет им. П. Г. Демидова

Аннотация. Рассматриваются алгоритмически неразрешимые проблемы для уравнений в свободных группах с «достаточно простыми» подгрупповыми ограничениями на решения, разрешенныхе относительно неизвестных с простой правой частью. Исследуются алгоритмически неразрешимые проблемы для уравнений в словах и длинах в свободных полугруппах с одним дополнительным ограничением на решение.

Ключевые слова: проблема совместности для систем уравнений, уравнения в свободных группах, уравнения в словах и длинах.

*Светлой памяти*
*Михаила Абрамовича Тайцлина*
*посвящается*

## Введение

Трудно отрицать, что на протяжении всей многовековой истории «Алгебры» изучение уравнений играло достаточно важную роль. Приведем цитату из монографии Ж. А. Серре «Курс высшей алгебры» (Русский перевод, СПб: 1910): «Алгебра — это, по сути, жонглирование уравнениями». Конечно, с начала 20-х годов прошлого века «Алгебра» — это наука об алгебраических операциях. «Алгебра» изучает алгебраические структуры или даже алгебраические системы.

Важную роль в истории «Алгебры» играет изучение уравнений и их систем с различными ограничениями на решения. Эта традиция прослеживается со времен Диофанта (предположительно III век

н. э.), который начал изучать уравнения с рациональными (положительными) ограничениями на решения, то есть в современных обозначениях системы вида

$$F(x_1, \ldots, x_n) = G(x_1, \ldots, x_n) \,\&\, \mathop{\&}_{i=1}^{n} x_i \in \mathbb{Q}_+,$$

где $F(x_1, \ldots, x_n)$ и $G(x_1, \ldots, x_n)$ — многочлены с положительными рациональными коэффициентами, то есть многочлены над множеством положительных рациональных чисел $\mathbb{Q}_+$. Диофанта интересовал, прежде всего, вопрос о нахождении какого-нибудь рационального решения конкретного уравнения и нахождение (получение, описание) всех его рациональных решений, отправляясь, как правило, от одного известного решения.

Вопрос, имеет ли рациональное решение рассматриваемое уравнение не обсуждался, в частности, из-за определенной простоты рассматривавшихся уравнений — они имели, как правило, степень 2 и лишь две неизвестные и одно решение обычно легко находилось.

Важный вклад в изучение уравнений с ограничениями на решения внес П. Ферма, который во «Втором вызове математикам» (британским) (1657 г.) рассматривает уравнение с ограничением на решение

$$ax^2 + 1 = y^2 \,\&\, x, y \in \mathbb{N}$$

и предлагает доказать, что оно всегда имеет (натуральное) решение, если натуральное число $a$ не является полным квадратом и найти какое-нибудь решение при нескольких указанных в «Вызове» значениях $a$, в частности, при $a = 149109433$.

Это уравнение теперь хорошо известно, с «легкой руки» Л. Эйлера оно называется уравнением Пелля и обычно записывается в виде

$$x^2 - ay^2 = 1.$$

С тех пор изучение уравнений и их систем с различными ограничениями на решения — важная задача «Теории чисел», «Алгебры» и «Теории алгоритмов».

Напомним, что 10-я проблема Д. Гильберта — это вопрос о существовании общего метода (алгоритма), позволяющего по произвольному уравнению с ограничениями на решения вида

$$F(x_1, \ldots, x_n) = 0 \,\&\, \mathop{\&}_{i=1}^{n} x_i \in \mathbb{Z},$$

где $F(x_1, \ldots, x_n)$ — многочлен над кольцом целых чисел $\mathbb{Z}$, определить, имеет ли оно решение.

Хорошо известно, что эта проблема решена отрицательно в работах М. Дэвиса, Дж. Робинсон, Х. Путнама и Ю. В. Матиясевича [20]. Отрицательно решается и равносильный предыдущему вопросу вопрос о существовании алгоритма, позволяющего по произвольному уравнению с ограничениями на решения вида

$$F(x_1, \ldots, x_n) \;=\; 0 \;\&\; \mathop{\&}_{i=1}^{n} x_i \in \mathbb{N},$$

где $\mathbb{N}$ — множество натуральных чисел, определить, имеет ли оно решение.

Вопрос о наличии алгоритма, решающего проблему существования решения для уравнений с ограничениями на решения вида

$$F(x_1, \ldots, x_n) \;=\; 0 \;\&\; \mathop{\&}_{i=1}^{n} x_i \in \mathbb{Q},$$

где $\mathbb{Q}$ — поле рациональных чисел, в настоящее время открыт.

Вопрос о наличии алгоритма, решающего проблему существования решения для уравнений с ограничениями на решения вида

$$F(x_1, \ldots, x_n) \;=\; 0 \;\&\; \mathop{\&}_{i=1}^{n} x_i \in \mathbb{R},$$

где $\mathbb{R}$ — поле действительных чисел, достаточно давно положительно решен А. Тарским, далеко обобщившим и расширившим метод Штурма для уравнений с одной неизвестной. Более того А. Тарский на этой основе разработал разрешающий алгоритм для элементарной теории поля действительных чисел, то есть алгоритм, позволяющий по произвольной замкнутой формуле $\Phi$ в сигнатуре $\langle 0, 1, +, \cdot, \leqslant \rangle$ определить, истинна ли она на поле действительных чисел. В этой связи необходимо указать на фундаментальный результат Ю. Л. Ершова, Дж. Акса и С. Кочена, которые доказали разрешимость элементарной теории любого поля $\mathbb{Q}_p$ $p$-адических чисел. Напомним, что поле $\mathbb{R}$ действительных чисел и каждое поле $\mathbb{Q}_p$ $p$-адических чисел являются пополнением поля $\mathbb{Q}$ рациональных чисел относительно соответствующей нормы.

Напомним, что проблема разрешимости для уравнений с ограничениями на решения вида

$$F(x_1, \ldots, x_n) \;=\; 0 \;\&\; \mathop{\&}_{i=1}^{n} x_i \in \{\, 0,\, 1 \,\}$$

является $NP$-полной.

В заключение напомним, что Д. Гильберт доказал следующую фундаментальную теорему:

Теорема 1. *Для произвольных многочленов $F_i(x_1, \ldots, x_n)$, $i \in I$, с комплексными коэффициентами система*

$$\mathop{\&}_{i\in I} F_i(x_1, \ldots, x_n) = 0 \mathop{\&}_{i=1}^{n} x_i \in \mathbb{C}$$

*не имеет решения, если и только если многочлены $F_i(x_1, \ldots, x_n)$, $i \in I$, порождают в кольце $\mathbb{C}[x_1, \ldots, x_n]$ единичный идеал, то есть существуют такие многочлены $G_i(x_1, \ldots, x_n)$, $i \in I$, с комплексными коэффициентами, для которых выполняется равенств*

$$\sum_{i\in I} G_i(x_1, \ldots, x_n) F_i(x_1, \ldots, x_n) = 1.$$

## 1. Основные определения

Через $M_m$ мы будем, как обычно, обозначать свободный моноид, то есть свободную полугруппу с пустым словом в качестве нейтрального элемента, ранга $m$ со свободными образующими $a_1, \ldots, a_m$, а через $F_m$ — свободную группу с теми же свободными образующими. Вместо $a_1$ и $a_2$ будем писать $a$ и $b$ соответственно.

Уточним некоторые определения, относящиеся к системам уравнений в свободных группах.

Определение 1 (Система уравнений в свободной группе). *Системой уравнений с неизвестными $x_1, \ldots, x_n$ в свободной группе $F_m$ называется выражение вида*

$$\mathop{\&}_{i=1}^{k} w_i(x_1, \ldots, x_n, a_1, \ldots, a_m) = u_i(x_1, \ldots, x_n, a_1, \ldots, a_m), \qquad (1)$$

*где $w_i(x_1, \ldots, x_n, a_1, \ldots, a_m)$ и $u_i(x_1, \ldots, x_n, a_1, \ldots, a_m)$ — слова в алфавите $\{ x_1, x_1^{-1}, \ldots, x_n, x_n^{-1}, a_1, a_1^{-1}, \ldots, a_m, a_m^{-1} \}$. Набор $\langle g_1, \ldots, g_n \rangle$ элементов группы $F_m$ называется р е ш е н и е м системы (1), если при любом $i = 1, \ldots, k$ в группе $F_m$ выполнено*

$$w_i(g_1, \ldots, g_n, a_1, \ldots, a_m) = u_i(g_1, \ldots, g_n, a_1, \ldots, a_m).$$

Две системы уравнений с одними и теми же неизвестными называются э к в и в а л е н т н ы м и, если множества их решений совпадают. Используя, например, уравнение

$$[x, a] = ([x, b]\, y^2)^2,$$

имеющее в свободной группе $F_m$ при любом $m \geqslant 2$ лишь тривиальное решение $x = 1$, $y = 1$, любую систему уравнений (1) можно заменить одним, ей равносильным, уравнением.

Определение 2 (Система уравнений в свободном моноиде, свободной полугруппе). *Системой уравнений с неизвестными* $x_1, \ldots, x_n$ *в свободном моноиде (свободной полугруппе)* $M_m$ *называется выражение вида*

$$\mathop{\&}_{i=1}^{k} w_i(x_1, \ldots, x_n, a_1, \ldots, a_m) = u_i(x_1, \ldots, x_n, a_1, \ldots, a_m), \qquad (2)$$

*где* $w_i(x_1, \ldots, x_n, a_1, \ldots, a_m)$ *и* $u_i(x_1, \ldots, x_n, a_1, \ldots, a_m)$ — *слова в алфавите* $\{ x_1, x_2, \ldots, x_n, a_1, a_2, \ldots, a_m \}$. *Набор* $\langle g_1, \ldots, g_n \rangle$ *элементов моноида* $M_m$ *называется решением системы* (2), *если при любом* $i = 1, \ldots, k$ *в моноиде* $M_m$ *выполнено*

$$w_i(g_1, \ldots, g_n, a_1, \ldots, a_m) = u_i(g_1, \ldots, g_n, a_1, \ldots, a_m).$$

Две системы уравнений с одними и теми же неизвестными называются эквивалентными, если множества их решений совпадают. При $m \geqslant 2$ система уравнений (2) равносильна одному уравнению

$$w_1 a_1 w_2 a_1 \ldots a_1 w_k \, w_1 a_2 w_2 a_2 \ldots a_2 w_k = \\ = u_1 a_1 u_2 a_1 \ldots a_1 u_k \, u_1 a_2 u_2 a_2 \ldots a_2 u_k.$$

## 2. Из истории обсуждаемого вопроса

Исследование разрешимости уравнений в свободных группах было начато в конце 50-х годов прошлого века в связи с проблемой разрешимости элементарных теорий свободных групп, поставленной А. Тарским [40]. Этому вопросу посвящены, в частности, работы [12,24,26,37]. В 60-е годы прошлого века А. А. Марков предложил использовать системы уравнений в свободном моноиде $M_m$ в качестве одного из подходов к отрицательному решению 10-й проблемы Д. Гильберта.

Системы уравнений в свободных моноидах (в свободных полугруппах) также называются системами уравнений в словах. Первые результаты в исследовании систем уравнений в словах были получены А. А. Марковым (не опубликовано) и Ю. И. Хмелевским [25] в конце 60-х годов прошлого века.

В эти же годы было начато изучение систем уравнений в словах и длинах, то есть систем вида

$$\mathop{\&}_{t=1}^{k} w_t(x_1, \ldots, x_n) = u_t(x_1, \ldots, x_n) \;\&\; \mathop{\&}_{\{i,j\} \in A} |x_i| = |x_j|,$$

где через $|x| = |y|$ обозначен предикат «длины слов $x$ и $y$ равны». Первые результаты в исследовании систем уравнений в словах и длинах были получены в начале 70-х годов в работах Ю. В. Матиясевича [21] и Н. К. Косовского [8–10].

В 1976 году Г. С. Маканин получил в теории уравнений в словах фундаментальный результат, который был опубликован в 1977 году в работах [13] и [14], — он построил алгоритм, позволяющий по произвольной системе уравнений в свободной полугруппе $M_m$ определить, имеет ли она решение. Несколько позже в работе [15] 1982 года Г. С. Маканин построил алгоритм, позволяющий по произвольной системе уравнений в свободной группе $F_m$ определить, имеет ли она решение — он построил такую рекурсивную функцию $\Phi(d)$, что если данное уравнение с длиной записи $d$ имеет решение в свободной группе, то длина каждой компоненты минимального (по максимальной длине компоненты) решения не превосходит числа $\Phi(d)$. Это дает переборный алгоритм для распознавания разрешимости произвольного уравнения в свободной группе.

В связи с уже упоминавшейся выше проблемой А. Тарского о разрешимости элементарной теории произвольной свободной группы представлял интерес вопрос об алгоритмической природе фрагментов этой теории. Основные результаты в этой области были получены Г. С. Маканиным — вскоре после опубликования работы [15] ему удалось на том же пути доказать разрешимость экзистенциональной (универсальной) и позитивной теорий любой свободной группы [16]. При доказательстве разрешимости позитивной теории свободной группы Г. С. Маканин использовал результат Ю. И. Мерзлякова [22] об устранимости кванторов общности в позитивных формулах, относящихся к свободным группам. Вопрос о разрешимости позитивной теории свободной группы был сведен Ю. И. Мерзляковым [22] к следующей проблеме для уравнений с ограничениями на решения:

*Существует ли алгоритм, который для произвольного уравнения*

$$w(x_1, \ldots, x_n, a_1, \ldots, a_m) = 1$$

*в свободной группе счетного ранга определяет, имеет ли оно такое*

*решение* $g_1, \ldots, g_n$, *что* $g_1 \in F_{m_1}, g_2 \in F_{m_2}, \ldots, g_n \in F_{m_n}$, *где* $m_1 \leqslant m_2 \leqslant \ldots \leqslant m_n$, $F_{m_i}$ — *свободная группа с образующими* $a_1, \ldots, a_{m_i}$.

Г. С. Маканин [16] построил искомый алгоритм, и тем самым доказал разрешимость позитивной теории свободной группы.

После фундаментальных результатов Г. С. Маканина особый интерес стал представлять вопрос о существовании аналогичных алгоритмов для уравнений в свободных моноидах, полугруппах и группах с различными «не слишком сложными» и «достаточно естественными» ограничениями на решения.

В работах [2, 3] первым автором была доказана алгоритмическая неразрешимость позитивной $\exists\forall\exists^3$-теории любой конечно порожденной нециклической свободной полугруппы. Вопрос о разрешимости позитивной теории свободной полугруппы счетного ранга в его кандидатской диссертации 1973 года был сведен путем переноса методов Ю. И. Мерзлякова [22] со свободных групп на свободную счетнопорожденную полугруппу к следующей проблеме:

*Существует ли алгоритм, который для произвольного уравнения*

$$w(x_1, \ldots, x_n, a_1, \ldots, a_m) = u(x_1, \ldots, x_n, a_1, \ldots, a_m)$$

*в свободной полугруппе счетного ранга говорит, имеет ли оно решение* $g_1, \ldots, g_n$, *где* $g_1 \in M_{m_1}, g_2 \in M_{m_2}, \ldots, g_n \in M_{m_n}$, $m_1 \leqslant m_2 \leqslant \ldots \leqslant m_n$, $M_{m_i}$ — *свободная полугруппа с образующими* $a_1, \ldots, a_{m_i}$.

На основе этого сведения в работе [5] на стр. 150 отмечается «Нетрудно показать, что позитивная теория свободной полугруппы $\Pi_\infty$ является рекурсивно перечислимой, но не известно, является ли она рекурсивной». Через $\Pi_\infty$ обозначается свободная полугруппа счетного ранга.

Ю. М. Важенин и Б. В. Розенблат [1], используя фундаментальный результат Г. С. Маканина [13], доказали, что для решения последней задачи алгоритм существует, это позволило им установить разрешимость позитивной теории свободной полугруппы счетного ранга.

Обобщая эти ситуации, Г. С. Маканин поставил в «Коуровской тетради» [11] такую проблему для уравнений в свободных группах:

**9.25.** *Указать алгоритм, который по уравнению*

$$w(x_1, \ldots, x_n, a_1, \ldots, a_m) = 1$$

*в свободной группе* $F_m$ *и списку конечно порожденных подгрупп* $H_1, \ldots, H_n$ *группы* $F_m$ *позволял бы узнать, существует*

*ли решение этого уравнения с условием* $x_1 \in H_1, \ldots, x_n \in H_n$.

Первые положительные результаты в направлении решения этой проблемы были получены А. Ш. Малхасяном [17].

К. Шульц [38] рассмотрел аналогичную сформулированной выше проблеме 9.25 Г. С. Маканина проблему для уравнений в свободных моноидах (свободных полугруппах) с регулярными ограничениями на решения и доказал следующее:

Теорема 2. *Существует алгоритм, который по уравнению*

$$w(x_1, \ldots, x_n, a_1, \ldots, a_m) = u(x_1, \ldots, x_n, a_1, \ldots, a_m)$$

*в свободном моноиде* $M_m$ *и списку регулярных подмножеств (языков)* $H_1, \ldots, H_n$ *моноида* $M_m$ *позволяет узнать, существует ли решение этого уравнения с условием* $x_1 \in H_1, \ldots, x_n \in H_n$.

Так как каждая конечно порожденная подполугруппа свободного моноида $M_m$ является регулярным подмножеством (языком), то решенная К. Шульцем проблема для уравнений с ограничениями на решения в свободных полугруппах является естественным аналогом проблемы Г. С. Маканина.

Ф. Дикерт [32–34] построил алгоритм, позволяющий по произвольному уравнению

$$w(x_1, \ldots, x_n, a_1, \ldots, a_m) = 1$$

в свободной группе $F_m$ и списку регулярных подмножеств (языков) $H_1, \ldots, H_n$ группы $F_m$ определить, существует ли решение этого уравнения с условием $x_1 \in H_1, \ldots, x_n \in H_n$. Тем самым решена и проблема 9.25 Г. С. Маканина. Более того, в указанных работах дана оценка емкостной сложности рассматриваемых проблем.

Сказанное дает основания считать, что представляет интерес дальнейшее исследование различных обобщений проблемы 9.25 Г. С. Маканина для свободных групп, моноидов и полугрупп, получающихся путем ослабления ограничений, налагаемых на подгруппы (подполугруппы, подмоноиды, языки) $H_1, \ldots, H_n$.

## 3. Основные результаты

Основные наши результаты, которые мы хотели бы обсудить, содержатся в следующих теоремах, доказательства которых базируются прежде всего на фундаментальной теореме М. Дэвиса,

Дж. Робинсон, Х. Путнам, Ю. В. Матиясевича о диофантовости любого рекурсивно перечислимого множества [20].

Теорема 3. *В свободной группе $F_2$ со свободными образующими $a$ и $b$ можно построить такое семейство разрешенных относительно неизвестных уравнений*

$$w(x^k, x_1, \ldots, x_n) = [a, b],$$

*где $w(x^k, x_1, \ldots, x_n)$ — групповое слово в алфавите неизвестных $x$, $x_1, x_2, \ldots, x_n$, что невозможен алгоритм, определяющий для произвольного натурального числа $k$, существует ли решение уравнения*

$$w(x^k, x_1, \ldots, x_n) = [a, b],$$

*где $x_1 \in F_2^{(1)}, \ldots, x_t \in F_2^{(1)}$, $t$ — некоторое фиксированное число между 1 и $n$, а $F_2^{(1)} = [F_2, F_2]$ — коммутант свободной группы $F_2$.*

Как обычно через $F_2^{(2)} = [F_2^{(1)}, F_2^{(1)}]$ обозначается второй коммутант свободной группы $F_2$.

Теорема 4. *Невозможен алгоритм, позволяющий по произвольному разрешенному относительно неизвестных уравнению вида*

$$w(x_1, \ldots, x_n) = [a, b]$$

*в свободной группе $F_2$ определить, имеет ли оно такое решение $g_1$, ..., $g_n$, что $g_1 \in F_2^{(2)}$.*

Заметим, что слово $[a, b]$, стоящее в правой части рассматриваемых в доказанной теореме уравнений, имеет длину 4. Следующая теорема показывает невозможность дальнейшего уменьшения длины правой части.

Теорема 5. *Существует полиномиальный алгоритм, позволяющий по произвольному разрешенному относительно неизвестных уравнению вида*

$$w(x_1, \ldots, x_n) = g(a, b),$$

*где $w(x_1, \ldots, x_n)$ является групповым словом в алфавите неизвестных $x_1, x_2, \ldots, x_n$, а $g(a, b)$ — элемент длины меньше 4 свободной группы $F_2$ со свободными образующими $a$ и $b$, определить, существует ли решение этого уравнения, для которого $x_1 \in F_2^{(s)}, \ldots, x_t \in F_2^{(s)}$, где $t$ — произвольное фиксированное число между 1 и $n$, а $F_2^{(s)}$ — $s$-й коммутант свободной группы $F_2$.*

Обозначим через $\varphi_i$ следующий эндоморфизм свободной группы

$F_m$ ранга $m$ со свободными образующими $a_1, \ldots, a_m$

$$\varphi_i(a_j) \rightleftharpoons a_j \text{ при } j \neq i, \quad \varphi_i(a_i) \rightleftharpoons 1.$$

По аналогии с группой кос эндоморфизм $\varphi_i$ назовем «эндоморфизмом выдергивания $i$-ой образующей».

Полагаем

$$P_m^{(i)} \rightleftharpoons \operatorname{Ker}\varphi_i, \quad P_m \rightleftharpoons \bigcap_{i=1}^{m} P_m^{(i)}$$

и назовем $P_m^{(i)}$ подгруппой $i$-чистых элементов, а $P_m$ — подгруппой чистых или гладких элементов.

Ясно, что $P_m$ — нормальная подгруппа группы $F_m$, содержащаяся в ее коммутанте $F_m^{(1)}$, $P_m \subseteq F_m^{(1)}$, и $P_2 = F_2^{(1)}$, но $P_m \neq F_m^{(1)}$ при $m \geqslant 3$.

Теорема 6. *При $m \geqslant 3$ невозможен алгоритм, позволяющий по произвольному уравнению в группе $F_m$*

$$w(x_1, \ldots, x_n, a_1, \ldots, a_m) = 1$$

*определить, имеет ли оно такое решение $x_1, \ldots, x_n$, что $x_1 \in P_m$.*

Следующую теорему можно рассматривать как усиление основного результата работы [31] о финитной неаппроксимируемости проблемы разрешимости уравнений в свободной группе.

Теорема 7. *При любом $n \geqslant 2$ и любых неотрицательных $m$, $p$ и $q$ уравнение*

$$((x^2u)^{2+p}(z^{-1}y^2vz)^{2+q}t^{2m+3})^4[u,v] = [a_1, a_2]$$

*не имеет решения в свободной группе $F_n$, однако уравнение*

$$((x^2u)^{2+p}(z^{-1}y^2vz)^{2+q}t^{2m+3})^4[u,v] = [\overline{a_1}, \overline{a_2}]$$

*имеет решение в любой конечной факторгруппе $F_n/N$, где через $\overline{a_1}$ и $\overline{a_2}$ обозначены образы свободных образующих $a_1$ и $a_2$ свободной группы $F_n$ относительно ее естественного гомоморфизма на факторгруппу $F_n/N$.*

В качестве некоторого дополнения к предыдущим теоремам может рассматриваться следующая теорема.

Теорема 8. *Проблема разрешимости в свободной группе $F_2$ для уравнений вида*

$$w(x_1, \ldots, x_n) = [a, b],$$

*где $w(x_1, \ldots, x_n)$ — слово в алфавите неизвестных, а $[a, b]$ — коммутатор свободных образующих $a$ и $b$ группы $F_2$ является $NP$-трудной.*

Заметим, что слово $[a, b]$ имеет длину 4. Для слов $g$ длины меньше 4 ситуация принципиально иная как показывает следующая теорема.

Теорема 9. *Проблема разрешимости для уравнений вида*

$$w(x_1, \ldots, x_n) = g,$$

*где $w(x_1, \ldots, x_n)$ является групповым словом в алфавите неизвестных $\{x_1, \ldots, x_n, \ldots\}$, а $g$ — групповое слово длины меньше 4 в алфавите $\{a, b\}$ свободных образующих группы $F_2$ полиномиально разрешима.*

Следующие теоремы относятся к уравнениям в словах и длинах с некоторыми дополнительными ограничениями. Уже более полувека остается открытым вопрос о существовании алгоритма, позволяющего по произвольной системе уравнений с ограничениями на решения в свободном моноиде $M_2$ вида

$$w\,(\,x,\,x_1,\ldots,x_n,\,a,\,b\,) = v\,(\,x,\,x_1,\ldots,\,x_n,\,a,\,b\,)\;\&\;\mathop{\&}_{\{i,j\}\,\in\,A}\,|x_i| = |x_j|$$

определить, имеет ли она решение.

В связи с этим представляют интерес следующие теоремы

Теорема 10. *Можно указать такое однопараметрическое семейство уравнений с ограничениями на решения в свободном моноиде $M_2$,*

$$w\,(\,x,\,x_1,\ldots,x_n,\,a,\,b\,) = v\,(\,x,\,x_1,\ldots,\,x_n,\,a,\,b\,)\;\&\;\mathop{\&}_{\{i,j\}\,\in\,A}\,|x_i| = |x_j|\;\&\;|x_1|_b = |x_2|_b$$

*с неизвестными $x_1, \ldots, x_n$, с константами $a$ и $b$ и с параметром $x$, где $A$ — некоторое подмножество множества $\{\{t, s\} \mid 1 \leqslant t, s \leqslant n\}$, что невозможен алгоритм, позволяющий для произвольного натурального числа $m$ определить, имеет ли решение уравнение с ограничениями на решения*

$$w\,(\,a^m,\,x_1,\ldots,x_n,\,a,\,b\,) = v\,(\,a^m,\,x_1,\ldots,\,x_n,\,a,\,b\,)\;\&\;\mathop{\&}_{\{i,j\}\,\in\,A}\,|x_i| = |x_j|\;\&\;|x_1|_b = |x_2|_b.$$

Теорема 11. *Можно указать такое однопараметрическое семейство уравнений с ограничениями на решения в свободном моноиде $M_2$,*

$$w\,(\,x,\,x_1,\ldots,x_n,\,a,\,b\,) = v\,(\,x,\,x_1,\ldots,\,x_n,\,a,\,b\,)\;\&\;\mathop{\&}_{\{i,j\}\,\in\,A}\,|x_i| = |x_j|\;\&\;x_1 \in L_1$$

*с неизвестными $x_1, \ldots, x_n$, с константами $a$ и $b$ и с параметром $x$, где $A$ — некоторое подмножество множества $\{\{t, s\} \mid 1 \leqslant t, s \leqslant$*

$n\}$, *что невозможен алгоритм, позволяющий для произвольного натурального числа* $m$ *определить, имеет ли решение уравнение с ограничениями на решения*

$$w(a^m, x_1, \ldots, x_n, a, b) = v(a^m, x_1, \ldots, x_n, a, b) \,\&\, \&\, \&_{\{i,j\} \in A} |x_i| = |x_j| \,\&\, x_1 \in L_1.$$

Ф. Дикерт предложил (устное сообщение Ю. В. Матиясевича) изучать в свободных моноидах системы вида

$$\&_{i=1}^{k} w_i(x_1, \ldots, x_n, a_1, \ldots, a_m) \leqslant u_i(x_1, \ldots, x_n, a_1, \ldots, a_m), \qquad (3)$$

где для слов $w$ и $u$ в алфавите образующих свободного моноида запись $w \leqslant u$ означает, что последовательность букв $w$ является подпоследовательностью букв $u$, то есть существуют такое число $n \leqslant |w|$ и такие слова $w_1, \ldots, w_k, u_1, \ldots, u_k, u_{k+1}$, что

$$w = w_1 \ldots w_k \quad u = u_1 w_1 u_2 \ldots u_k w_k u_{k+1},$$

рассматривая их как обобщение систем уравнений (1), так как $w = u$ тогда и только тогда, когда $w \leqslant u \,\&\, u \leqslant w$.

Отношение $w \leqslant u$ является отношением частичного порядка на моноиде $M_m$, то есть оно рефлексивно, транзитивно и антисимметрично. Это еще один довод для обоснования естественности рассмотрения систем неравенств вида (1).

Вопрос об алгоритмической разрешимости проблемы совместности для систем неравенств (3) в настоящее время открыт. Но если к отношению $w \leqslant u$ добавить предикат равенства длин, то получим алгоритмически неразрешимую задачу.

В дальнейшем равенство $w = u$ будет использоваться как сокращенная запись конъюнкции неравенств $w \leqslant u \,\&\, u \leqslant w$.

Теорема 12. *Невозможен алгоритм, позволяющий для произвольной системы вида*

$$\&_{i=1}^{k} w_i \leqslant u_i \,\&\, \&_{\{i,j\} \in A} |x_i| = |x_j|$$

*определить, имеет ли она решение.*

## 4. Заключение

Обсуждаемые в докладе вопросы, на наш взгляд, вписываются в две фундаментальные проблемы, которые в 60-е годы прошлого века сформулировал Сергей Иванович Адян:

- Где «проходит граница» между областями «Алгоритмически разрешимые проблемы» и «Алгоритмически неразрешимые проблемы»?

- Какие дополнительные условия «превращают» «Алгоритмически разрешимую проблему» в «Алгоритмически неразрешимую проблему»?

## Список литературы

[1] *Важенин, Ю. М.* Разрешимость позитивной теории свободной счетнопорожденной полугруппы / Ю. М. Важенин, Б. В. Розенблат // Математический сборник. — 1981. — Т. 116, № 1. — С. 120–127.

[2] *Дурнев, В. Г.* Позитивная теория свободной полугруппы // Доклады АН СССР. — 1973. — Т. 211, № 4. — С. 772–774.

[3] *Дурнев, В. Г.* О позитивных формулах на свободных полугруппах // Сибирский математический журнал. — 1974. — Т. 25, № 5. — С. 1131–1137.

[4] *Дурнев, В. Г.* Неразрешимость позитивной $\forall\exists^3$-теории свободной полугруппы // Сибирский математический журнал. — 1995. — Т. 36, № 5. — С. 1067–1080.

[5] *Дурнев, В. Г.* О позитивной теории свободной полугруппы // Сб. «Вопросы теории групп и полугрупп». — Тула : Тульский государственный педагогический институт им. Л. Н. Толстого. — 1972. — С. 122–172.

[6] *Дурнев, В. Г.* Об уравнениях на свободных полугруппах и группах // Математические заметки. — 1974. — Т. 16, № 5. — С. 717–724.

[7] *Дурнев, В. Г.* О проблеме разрешимости для уравнений с одним коэффициентом // Математические заметки. — 1996. — Т. 59, № 6. — С. 832–845.

[8] *Косовский, Н. К.* Некоторые свойства решений уравнений в свободной полугруппе // Записки научных семинаров Ленинградского отделения Математического института АН СССР. Ленинград. — 1972. — Т. 32. — С. 21–28.

[9] *Косовский, Н. К.* О множествах, представимых в виде решений уравнений в словах и длинах // Вторая всесоюзная конференция по математической логике. Тезисы кратких сообщений. — М . — 1972. — С. 23.

[10] *Косовский, Н. К.* О решении систем, состоящих одновременно из уравнений в словах и неравенств в длинах слов // Записки научных семинаров Ленинградского отделения Математического института АН СССР. Ленинград. — 1973. — Т. 33. — С. 24–29.

[11] Коуровская тетрадь. Издание 17-е, дополненное, включающее Архив решенных задач / Сост. В. Д. Мазуров, Е. И. Хухро. — Новосибирск : Институт математики СО РАН, 2010. — 219 с.

[12] *Лоренц, А. А.* О представлении множеств решений систем уравнений с одним неизвестным в свободных группах // Доклады АН СССР. — 1968. — Т. 178, № 2. — С. 290–292.

[13] *Маканин, Г. С.* Проблема разрешимости уравнений в свободной полугруппе// Доклады АН СССР. — 1977. — Т. 233, № 2. — С. 287–290.

[14] *Маканин, Г. С.* Проблема разрешимости уравнений в свободной полугруппе // Математический сборник. — 1977. — Т. 103 (145), № 2 (6). — С. 147–236.

[15] *Маканин, Г. С.* Уравнения в свободной группе // Известия АН СССР. Серия математика. — 1982. — Т. 46, № 6. — С. 1199–1273.

[16] *Маканин, Г. С.* Универсальная теория и позитивная теория свободной группы // Известия АН СССР. Серия математика. — 1984. — Т. 48, № 4. — С. 735–749.

[17] *Малхасян, А. Ш.* О разрешимости в подгруппах уравнений в свободной группе // Сборник «Прикладная математика». — 1986. — Т. 2. — С. 42–47.

[18] *Мальцев, А. И.* Об уравнении $zxyx^{-1}y^{-1}z^{-1} = aba^{-1}b^{-1}$ в свободной группе // Алгебра и логика. — 1962. — Т. 1, № 5. — С. 45–50.

[19] *Мальцев, А. И.* О гомоморфизмах на конечные группы // Ученые записки Ивановского педагогического института. — 1958. — Т. 18. — С. 49–60.

[20] *Матиясевич, Ю. В.* Диофантовость перечислимых множеств // Доклады АН СССР. — 1970. — Т. 130, № 3. — С. 495–498.

[21] *Матиясевич, Ю. В.* Связь систем уравнений в словах и длинах с 10-ой проблемой Гильберта // Исследования по конструктивной математике и математической логике. Записки научных семинаров Ленинградского отделения Математического института АН СССР. Ленинград. — 1968. — Т. 8. — С. 132–143.

[22] *Мерзляков, Ю. И.* Позитивные формулы на свободных группах // Алгебра и логика. — 1966. — Т. 5, № 4. — С. 25–42.

[23] *Разборов, А. А.* О системах уравнений в свободной группе // Известия АН СССР. Серия математика. — 1984. — Т. 48, № 4. — С. 779–832.

[24] *Хмелевский, Ю. И.* Системы уравнений в свободной группе. I, II. // Известия АН СССР. Серия математика. — 1971. — Т. 35, № 6. — С. 1237–1268. — 1972. — Т. 36, № 1. — С. 110–179.

[25] *Хмелевский, Ю. И.* Уравнения в свободной полугруппе. — М. : Наука. — 1971. (Тр. МИАН.) Т. 107).

[26] *Appel, K. I.* One-variable equations in free groups // Proceedings of the American Mathematical Society. — 1968. — Vol. 19. — P. 912–918.

[27] *Baumslag, G.* Residual nilpotency and relations in free groups // Journal of Algebra. — 1965. — Vol. 2. — P. 271–282.

[28] *Birman, J. S.* Braids, links and mapping class groups. — Princeton, New Jersey : Princeton University Press, 1974.

[29] *Büchi, J. R.* Definability in the existential theory of concatenation / J. R. Buchi, S. Senger // Zeitschrift für mathematische Logik und Grundlagen der Mathematik — 1988. — V. 34, № 4. — P. 337–342.

[30] *Büchi, J. R.* Coding in the existential theory of concatenation / J. R. Buchi, S. Senger // Archive for Mathematical Logic. — 1986/87. Bd. 26. — P. 101–106.

[31] *Coulbois, T.* Equations in free groups are not finitely approximable / T. Coulbois, A. Khelif // Proceedings of the American mathemetical society. — 1999. — V. 127, № 4. — P. 963–965.

[32] *Diekert, V.* Makanin's Algorithm for Solving Word Equations with Regular Constraints. Preliminary version of the chapter in M. Lothaire. Algebraic Combinatorics on Words. Report Nr. 1998/02. Fakultat Informatik. Universitat Stuttgart. — 1998.

[33] *Diekert, V.* The existential theory of equations with rational constraints in free groups is PSPACE-complete / V. Diekert, C. Gutierrez, C. Hagenah // In A. Ferreira and H Reichel, editors, Proc. 18-th Annual Symposium on Theoretical Aspects of Computer Science (STACS'01), Dresden (Germany). — 2000, Vol. 2010 in Lecture Notes in Computer Science. — Berlin, Heidelberg : Springer-Verlag, 2001. — P. 170–182.

[34] *Diekert, V.* The existential theory of equations with rational constraints in free groups is PSPACE-complete / V. Diekert, C. Gutierrez, C. Hagenah // Informatiion and Computation. — 2005. — Vol. 202. — P. 105–140.

[35] *Edmunds, C. C.* On the endomorphisms problem for free group // Communications in Algebra. — 1975. — Vol. 3. — P. 7–20.

[36] *Gassner, B. J.* On braid groups // Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg. — 1961. — Vol. 25. — P. 10–22.

[37] *Lyndon, R. C.* Equations in free groups // Transactions of the American Mathematical Society. — 1960. — Vol. 96. — P. 445–457.

[38] *Schulz, K. U.* Makanin's Algorithm for Word Equations - Two Improvements and a Generalization // Lecture Notes in Computer Science. — 1990. — Vol. 572. — P. 85–150.

[39] *Schupp, P. E.* On the substitution problem for free groups // Proceedings of the American Mathematical Society. — 1969. — Vol. 23. — P. 421–423.

[40] *Tarski, A.* Undecidable theories / A. Tarski, A. Mostowski, R. M. Robinson. — NY. — 1953.

### Библиографическая ссылка

*Дурнев, В. Г.* Алгоритмические проблемы для уравнений в свободных группах и полугруппах с ограничениями на решения / В. Г. Дурнев, А. И. Зеткина // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 25–41.
https://doi.org/10.26456/mfcsics-21-3

### Сведения об авторах

1. Валерий Георгиевич Дурнев
Ярославский государственный университет им. П. Г. Демидова. Профессор кафедры компьютерной безопасности и математических методов обработки информации

*Россия, 150000, Ярославль, ул. Советская, 14/2, ЯрГУ*
*E-mail: durnev@uniyar.ac.ru*

2. Алена Игоревна Зеткина
Ярославский государственный университет им. П. Г. Демидова. Аспирант

*Россия, 150000, Ярославль, ул. Советская, 14/2, ЯрГУ*
*E-mail: a.zetkina1@uniyar.ac.ru*

УДК 519.681.2
AMS MSC2020: 68Q85

# О системах переписывания процессов высокого уровня

**Ломазова И. А.**

НИУ «Высшая школа экономики»

Аннотация. Системы переписывания процессов (Process Rewrite Systems — PRS) Ричарда Майра представляют собой систему переписывания термов специального вида и задают унифицированное представление для конечных и магазинных автоматов, сетей Петри и некоторых классов алгебр процессов. В докладе рассматривается (P,P)-подкласс систем переписывания процессов, соответствующий классическим сетям Петри, и его расширение HPRS для моделирования систем с динамической структурой. Обсуждаются вопросы выразительности и разрешимости.

Ключевые слова: модели распределенных вычислений, системы переписывания процессов, выразительность, разрешимость.

Формальные модели распределенных систем остаются актуальной темой для исследования в связи с их важностью для приложений, например, для моделирования и анализа поведения сетевых агентов, а также распределенных информационных и технологических систем. При этом очень актуальной является разработка формализмов для моделирования сложных систем с динамической структурой, в частности, для моделирования адаптивных систем.

Системы переписывания процессов (Process Rewrite Systems — PRS) Ричарда Майра [2] представляют собой систему переписывания термов специального вида и задают унифицированное представление для конечных и магазинных автоматов, сетей Петри и некоторых классов алгебр процессов.

Пусть $Act = \{a, b, \dots\}$ — бесконечное множество имен действий, $Atom = \{\varepsilon, A, B, \dots\}$ — бесконечное множество атомов. В PRS-системах термы строятся из атомов с помощью операций параллельной (обозначается $\_\|\_$) и последовательной композиции (обозначается $\_.\_$). При этом считается, что операция параллельной

композиции ассоциативна и коммутативна, операция последовательной композиции ассоциативна.

PRS-система определяется как пара $(t_0, \Delta)$, где $t_0$ — терм, задающий начальное состояние, $\Delta$ — множество правил (подстановок) вида $(t_1 \xrightarrow{a} t_2)$, где $t_1$ и $t_2$ — термы, $a$ — имя действия из $Act$.

(P,P)-подкласс систем переписывания процессов, когда в термах используется только операция параллельной композиции, соответствует классическим сетям Петри.

Отношение перехода $\xrightarrow{a}$, где $a \in Act$, задается тогда правилами вывода:

$$\mathbf{R_1}: \frac{(t_1 \xrightarrow{a} t_2) \in \Delta}{(t_1 \xrightarrow{a} t_2)} \qquad \mathbf{R_2}: \frac{t_1 \xrightarrow{a} t_1'}{t_1 \| t_2 \xrightarrow{a} t_1' \| t_2} \qquad \mathbf{R_3}: \frac{t_2 \xrightarrow{a} t_2'}{t_1 \| t_2 \xrightarrow{a} t_1 \| t_2'}$$

Соответственно, семантика PRS-системы задается с помощью системы помеченных переходов, в которой состояния представлены PRS-термами, а переходы помечены именами действий.

В докладе рассматривается расширение (P,P)-подкласса систем переписывания процессов для моделирования систем с динамической структурой — системы переписывания процессов высокого уровня (HPRS-системы) [1].

В HPRS-системах термы наряду с атомами содержат переменные и подтермы вида $(t_1 \xrightarrow{a} t_2)$ — процедура $a$, соответствующая замене подтерма $t_1$ на $t_2$.

Конкретизацией процесса $P$ называется процесс $P'$, полученный из $P$ одновременной подстановкой термов вместо некоторых переменных в $P$ (обозначение: $P' \sqsubseteq_c P$)

Отношение перехода в HPRS определяется правилами вывода $\mathbf{R_1}, \mathbf{R_2}, \mathbf{R_3}$, а также правилом вызова процедуры:

$$\mathbf{R_4}: \frac{(P_1 \overset{a}{\longmapsto} P_2) \sqsubseteq_c (P_1' \overset{a}{\longmapsto} P_2')}{((P_1 \overset{a}{\longmapsto} P_2) \| P_1') \xrightarrow{a} ((P_1 \overset{a}{\longmapsto} P_2) \| P_2')}$$

В докладе обсуждаются вопросы выразительности HPRS-систем и разрешимости для них некоторых семантических свойств.

Заметим, что HPRS-системы действительно позволяют гибко менять структуру системы. Например, правило $X \| X \to X$ удаляет все кратные вхождения атомов в терме. Правило $(X \longmapsto Y) \to e$

удаляет из программы описания процедур. Правило $(X \longmapsto Y) \to X$ извлекает формальный параметр процедуры.

### Список литературы

[1] *Ломазова, И. А.* Универсальные сети Петри и системы переписывания процессов, дополненные процедурами // Доклады Академии наук. — 2005. — Т. 401, № 1. — С. 30–33.

[2] *Mayr, R.* Process Rewrite Systems // Information and Computation. — 2000. — Vol. 156, № 1. — P. 264–286.

### Библиографическая ссылка

*Ломазова, И. А.* О системах переписывания процессов высокого уровня // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 42–44.
https://doi.org/10.26456/mfcsics-21-4

### Сведения об авторах

Ирина Александровна Ломазова
НИУ «Высшая школа экономики». Профессор

*Россия, 109028, Москва, Покровский бульвар, 11, НИУ ВШЭ*
*E-mail: ilomazova@hse.ru*

УДК 510.6, 510.64
AMS MSC2020: 03B45

# Разрешимые свойства логик[1]

**Максимова Л. Л., Юн В. Ф.**

Институт математики им. С. Л. Соболева СО РАН

Аннотация. Рассматриваются логики над минимальной логикой $J$ и модальными логиками. Изучаются проблема узнаваемости, свойства табличности и предтабличности, различные интерполяционные свойства в классах расширений логики $J$ и модальных логик.

Ключевые слова: минимальная логика, модальная логика, разрешимость, алгоритмические свойства, узнаваемость, различимость, табличность, интерполяционные свойства.

## Введение

Рассматриваются логики над минимальной логикой $J$ и модальными логиками.

Свойство логик $P$ разрешимо над логикой $L_0$, если существует алгоритм, проверяющий по любой формуле $A$, обладает ли логика $L_0 + A$ свойством $P$; свойство $P$ сильно разрешимо над $L_0$, если такой алгоритм существует для всех конечных множеств $Rul$, составленных из схем аксиом и правил вывода.

Изучаются проблема узнаваемости, свойства табличности и предтабличности, различные интерполяционные свойства в классах расширений логики $J$ и модальных логик.

## 1. Различимость и узнаваемость

Пусть $L_0$ — логика, $L$ — конечно аксиоматизируемая логика, содержащая $L_0$. Говорим, что $L$ различима над $L_0$, если су-

[1]Работа выполнена в рамках государственного задания ИМ СО РАН (проект № 0314-2019-0002.)

ществует алгоритм, проверяющий по любой формуле $A$, верно ли включение $L_0 + A \geqslant L$.

Логика $L$ сильно различима над $L_0$, если существует алгоритм, проверяющий соотношение $L_0 + Rul \geqslant L$ для любого конечного множества $Rul$ аксиом и правил вывода. Логика $L$ узнаваема над $L_0$, если существует алгоритм, проверяющий по любой формуле $A$, верно ли равенство $L_0 + A = L$. Логика $L$ сильно узнаваема над $L_0$, если существует алгоритм, распознающий совпадение $L$ с $L_0 + Rul$.

Заметим, что если свойство $P$ разрешимо над $L_0$, то любая логика над $L_0$ со свойством $P$ узнаваема над $L_0$. Если свойство $P$ сильно разрешимо над $L_0$, то любая логика над $L_0$ со свойством $P$ сильно узнаваема над $L_0$.

Лемма 1 (см. [8, 11]). 1) *Логика $L$ узнаваема над $L_0$ тогда и только тогда, когда $L$ различима над $L_0$ и разрешима.*

2) *Если $L$ сильно различима над $L_0$ и проблема допустимости правил в $L$ разрешима, то $L$ сильно узнаваема над $L_0$. Из последнего следует, что $L$ различима над $L_0$ и разрешима по допустимости.*

Доказано, что логика $Int$ узнаваема над $J$ [8], логика $Neg = J + \bot$ сильно узнаваема над $J$ [11].

Неизвестно, является ли $Int$ сильно узнаваемой над $J$.

## 2. Интерполяционные свойства

Рассмотрим интерполяционное свойство Крейга $CIP$:

Если $L \vdash A \to B$, то существует $C$, содержащая лишь общие переменные $A$ и $B$ и такая, что $L \vdash A \to C$ и $L \vdash C \to B$, и другие варианты этого свойства.

Факты:

1) Слабое интерполяционное свойство $WIP$ разрешимо над $J$ [5], $wK4 = K + \{p \,\&\, \Box p \to \Box\Box p\}$ (Karpenko, 2012). Существует континуум слабо транзитивных модальных логик с $WIP$ и континуум $J$-логик с $WIP$.

2) Существует лишь конечное число логик над $S4$ с $IPR$, все логики над $S4$ с $IPR$ узнаваемы над $S4$. Интерполяционные

свойства $CIP$, $IPR$, проективное свойство Бета $PBP$ разрешимы над $S4$.

3) Проблема интерполяции полностью решена над $Int$. Все с. и. л. с $CIP$, $IPR$ и $PBP$ полностью описаны. Каждая из них узнаваема и даже сильно узнаваема над $Int$ [3, 12]. Существует точно восемь с. и. л. с $CIP$. Все с. и. л. имеют $WIP$.

Описаны все стройные логики, то есть логики над $J + \{(p \to q) \vee (q \to p)\}$ со свойствами $CIP$, $IPR$, $PBP$; получены другие технически сложные результаты. Однако неизвестно, конечно или бесконечно число $J$-логик со свойством $CIP$, и проблема интерполяции в расширениях минимальной логики $J$ еще далека от своего решения.

### 3. Табличность и предтабличность над $J$ и $S4$

Логика т а б л и ч н а, если характеризуется конечной алгеброй; п р е д т а б л и ч н а, если не является табличной, но любое ее расширение таблично.

Теорема 2. *Табличность и предтабличность разрешимы над $J$ и $S4$.*

Факты:

1) Пусть $L$ — модальная логика или $J$-логика. Логика $L$ таблична тогда и только тогда, когда $L$ не содержится ни в одной из предтабличных логик.

2) Существует точно три предтабличных с. и. л. [1], пять предтабличных расширений логики $S4$ ( [2], Esakia-Meskhi, 1977), семь предтабличных $J$-логик [9].

3) Все указанные предтабличные логики узнаваемы в соответствующих областях.

Существует континуум предтабличных логик над модальной логикой $K4$ (W. Blok, 1980); проблема табличности неразрешима над $K4$ (A. Chagrov).

**Список литературы**

[1] *Максимова, Л. Л.* Предтабличные суперинтуиционистские логики // Алгебра и логика. — 1972. — Т. 11, № 5. — С. 558–570.

[2] *Максимова, Л. Л.* Предтабличные расширения логики $S4$ Льюиса // Алгебра и логика. — 1975. — Т. 14, № 1 — С. 28–55.

[3] *Максимова, Л. Л.* Теорема Крейга в суперинтуиционистских логиках и амальгамируемые многообразия // Алгебра и логика. — 1977. — Т. 16, № 6. — С. 643–681.

[4] *Максимова, Л. Л.* Интерполяционные теоремы в модальных логиках и амальгамируемые многообразия топобулевых алгебр // Алгебра и логика. — 1979. — Т. 18, № 5. — С. 556–586.

[5] *Максимова, Л. Л.* Разрешимость слабого интерполяционного свойства над минимальной логикой // Алгебра и логика. — 2011. — Т. 50, № 2. — С. 152–188.

[6] *Максимова, Л. Л.* Разрешимость интерполяционного свойства Крейга в стройных $J$-логиках // Сибирский математический журнал. — 2012. — Т. 53, № 5. — С. 1048–1064.

[7] *Максимова, Л. Л.* Ограниченная интерполяция над модальной логикой $S4$ // Алгебра и логика. — 2013. — Т. 52, № 4. — С. 461–501.

[8] *Максимова, Л. Л.* Узнаваемые логики / Л. Л. Максимова, В. Ф. Юн // Алгебра и логика. — 2015, Т. 54, № 2. — С. 252–274.

[9] *Максимова, Л. Л.* Проблема табличности над минимальной логикой / Л. Л. Максимова, В. Ф. Юн // Сибирский математический журнал. — 2016. — Т. 57, № 6. — С. 1320–1332.

[10] *Максимова, Л. Л.* Узнаваемые и различимые логики и многообразия // Алгебра и логика. — 2017. — Т. 56, № 3. — С. 567–574.

[11] *Максимова, Л. Л.* Сильная разрешимость и сильная узнаваемость / Л. Л. Максимова, В. Ф. Юн // Алгебра и логика. — 2017. — Т. 56, № 5. — С. 559–581.

[12] *Maksimova, L.* Strongly decidable properties of modal and intuitionistic calculi. // Logic Journal of the IGPL. — 2000. — Т. 8, № 6. — С. 797–819.

### Библиографическая ссылка

*Максимова, Л. Л.* Разрешимые свойства логик / Л. Л. Максимова, В. Ф. Юн // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 45–49.
https://doi.org/10.26456/mfcsics-21-5

### Сведения об авторах

1. Максимова Лариса Львовна
Институт математики им. С. Л. Соболева СО РАН. Главный научный сотрудник

*630090, Новосибирск, пр. Академика Коптюга, 4*
*E-mail: lmaksi@math.nsc.ru*

2. Юн Вета Федоровна
Институт математики им. С. Л. Соболева СО РАН. Старший научный сотрудник

*630090, Новосибирск, пр. Академика Коптюга, 4*
*E-mail: yun@math.nsc.ru*

УДК 512.567, 004.825
AMS MSC2020: 08A72

# Основанные на решетках алгебраические системы и их приложения в задачах управления знаниями[1]

**Махортов С. Д.**

Воронежский государственный университет

Аннотация. Алгебраические методы предоставляют эффективные средства формального построения и исследования моделей в различных областях информатики, включая интеллектуальные системы. Это положение в полной мере относится к логическим системам продукционного типа, которые образуют теоретическую основу важного направления в дисциплине искусственного интеллекта. В докладе исследуется алгебраическая структура, моделирующая распределенную интеллектуальную систему продукционного типа с нечеткими правилами (FDLP-структура). Вводится класс уравнений, служащий теоретической основой нового метода обратного логического вывода. Метод направлен на минимизацию числа обращений к внешним источникам информации (FDLP-вывод). Обсуждаются стратегии обратного вывода, использующие несколько параметров релевантности. Рассматриваются вопросы компьютерной реализации FDLP-структур.

Ключевые слова: алгебраическая система, FDLP-структура, продукционно-логическое уравнение, нечеткая распределенная продукционная система, релевантный обратный вывод, компьютерная реализация.

## Введение

В последнее десятилетие автор разрабатывает алгебраическую теорию LP-структур (lattice production structures) [1], которая эффективно решает задачи, связанные с интеллектуальными системами

[1]Работа выполнена при финансовой поддержке РФФИ в рамках научного проекта № 19-07-00037.

продукционного типа. Один из важных разделов теории составляет метод релевантного обратного логического вывода (LP-вывод), направленный на снижение числа обращений к внешним источникам информации. Его теоретической основой служит аппарат продукционно-логических уравнений.

Существенной особенностью современных интеллектуальных систем является нечеткий характер знаний и рассуждений. Отсюда возникает задача распространения теории LP-структур на нечеткие продукционные системы. В то же время прогресс информационных технологий приводит к значительному увеличению масштабов и сложности решаемых задач. Компьютерные системы становятся большими и распределенными, что также требует продвижений в методах их построения. В ряде работ (в частности, [4]) было представлено соответствующее обобщение концепции LP-структур — определена алгебраическая модель, охватывающая нечеткие распределенные продукционно-логические системы.

Настоящая работа развивает алгебраическую модель исследованием класса продукционно-логических уравнений в FDLP-структуре. Процесс решения уравнения моделирует нечеткий обратный логический вывод в распределенной интеллектуальной системе. Обсуждаются новые стратегии обратного вывода с учетом нескольких параметров релевантности. Затрагиваются вопросы компьютерной реализации FDLP-структур.

Следует отметить, что известен ряд других интересных алгебраических подходов к формальному построению и исследованию интеллектуальных систем. К таковым, в частности, относится formal concept analysis (FCA) [2]. Он так же основан на решетках и рассматривает бинарные отношения между множествами. Однако предмет исследований настоящей работы и представленные здесь результаты не имеют пересечений с направлением FCA и другими известными альтернативами.

## 1. Основные понятия теории LP-структур

Вводится определение нечеткой LP-структуры. Это атомно-порожденная решетка, на которой задано (вторичное) нечеткое бинарное отношение $R$. В целях моделирования логического вывода

используется композиция отношений в классической семантике — (max-min)-композиция.

Отношение на решетке называется логическим, если обладает рядом дополнительных свойств: рефлексивность, транзитивность, дистрибутивность. Для произвольного отношения R может быть рассмотрено его логическое замыкание $\bar{R}$. Алгебраическая система, образованная решеткой и заданным на ней нечетким логическим отношением $\bar{R}$, называется FLP-структурой (fuzzy lattice production structure).

Рассматриваемый подход к моделированию и исследованию продукционных систем основан на представлении множеств фактов и правил FLP-структурой. Каждый элементарный факт отображается атомом решетки, предпосылка и заключение правила — соответствующими элементами решетки, а правила представляются нечетким бинарным отношением $R$, то есть его функцией принадлежности. Свойства отношения $\bar{R}$ отражают возможности логического вывода в моделируемой базе знаний.

Далее на FLP-структуре задается отображение в булеан, порождаемый множеством узлов некоторой вычислительной сети. Таким путем формируется расширенная алгебраическая система (FDLP-структура), моделирующая нечеткую распределенную базу знаний. На основе этой системы рассматриваются формальные признаки «хорошего» распределения информации по узлам сети.

## 2. Продукционно-логические уравнения в FDLP-структуре

Вводится связанный с FDLP-структурами класс уравнений. Правая часть уравнения — это заданный элемент решетки («гипотеза»). Решением называется любой минимальный прообраз гипотезы при отношении $\bar{R}$, принадлежащий так называемому начальному множеству решетки. Процесс нахождения решения уравнения моделирует обратный логический вывод в распределенной нечеткой базе знаний.

Основной источник трудностей для процесса решения состоит в том, что обычно задано лишь отношение $R$. Оно в моделируемой продукционной системе соответствует известному множеству продукций — базе знаний. Решение же требуется найти как прообраз гипотезы при отношении $\bar{R}$ — логическом замыкании $R$. При

этом полное построение логического замыкания нецелесообразно, поскольку на практике требует чрезмерный объем ресурсов. Другим фактором, усложняющим не только метод решения, но и саму постановку задачи, является нечеткость отношения $R$.

В представленных исследованиях получены результаты о разрешимости уравнений в нечеткой FDLP-структуре, а также обоснованы способы их решения. Приводятся формулировки основных теорем.

Аппарат уравнений в FDLP-структуре создает основу для продвижений в области оптимизации нечеткого распределенного логического вывода и верификации соответствующих баз знаний.

## 3. О применениях уравнений в FDLP-структурах

Излагаются идеи практического применения аппарата продукционно-логических уравнений для оптимизации нечеткого распределенного обратного вывода (FDLP-вывода) в интеллектуальных системах.

Стратегия направлена на минимизацию количества медленно выполняемых запросов (к базе данных или пользователю). Запросы по возможности должны соответствовать лишь тем фактам, которые действительно необходимы при выводе. Кроме того, при LP-выводе предпочтение отдается тестированию множеств фактов минимальной мощности.

Первая стадия обратного вывода на основе уравнения состоит в его решении — построении всех минимальных начальных прообразов гипотезы в FDLP-структуре. Далее в построенном множестве достаточно найти любой прообраз, атомы которого отображают лишь истинные факты, после чего сразу можно сделать заключение об истинности гипотезы.

Эффективный способ состоит в приоритетном просмотре прообразов, содержащих наиболее «релевантные» атомы решетки. Таковы в первую очередь атомы, присутствующие в максимальном количестве построенных прообразов. Тогда единственный отрицательный ответ на запрос исключает из рассмотрения сразу большое количество прообразов, что соответственно ускоряет исследование. Второй показатель релевантности атома — его присутствие в прообразах минимальной мощности. Таким образом, предпочтение отдается

тем прообразам, проверка истинности которых потребует меньшего количества вопросов пользователю (или обращений к базе данных).

Нечеткий и распределенный характер базы знаний создает дополнительные возможности (и трудности) для повышения эффективности логического вывода.

К задачам моделирования нечеткой продукционной системы относится достижение для доказываемой гипотезы более высокого значения коэффициента уверенности за приемлемое время. Таким образом, при формировании стратегий нечеткого LP-вывода необходимо учитывать еще один показатель релевантности атомов решетки. Он представляет значение функции принадлежности, вычисляемое для каждого найденного решения уравнения.

Еще одна очевидная цель моделирования интеллектуальной системы рассматриваемого типа — снижение трафика между узлами вычислительной сети в процессе выполнения логического вывода. Стратегия FDLP-вывода, наряду с упомянутыми выше тремя параметрами релевантности, должна использовать дополнительные показатели, связанные с атрибутами хранения фактов и правил на узлах сети.

Простейший вариант алгоритма определения релевантных атомов решетки в процессе FDLP-вывода состоит в элементарном суммировании значений используемых показателей. При этом имеется широкое поле для практических экспериментов, комбинирующих параметры релевантности с «весами», выбираемыми в зависимости от задач в конкретной предметной области. В будущем возможны более глубокие теоретические обоснования стратегий подсчета релевантности атомов, например, на основе методов многокритериальной оптимизации.

Компьютерная реализация FDLP-структур может основываться на представлении решеток битовыми векторами [3]. Разработана программная библиотека FDLPStructure, реализующая функциональные возможности новой теории. Эксперименты показывают, что при применении FDLP-вывода снижение числа выполняемых внешних запросов достигает в среднем 20–25%.

## Заключение

В представленной работе определен класс уравнений в FDLP-структуре, расширяющей границы применения алгебраической теории до нечетких распределенных интеллектуальных систем продукционного типа.

Установлены теоретические результаты, обосновывающие разрешимость таких уравнений и методы их решения. Процесс решения уравнения моделирует обратный нечеткий логический вывод в распределенной интеллектуальной системе. Таким образом, сформирована теоретическая база для оптимизации логического вывода.

На основе теории предложена концепция FDLP-вывода. Кратко описаны стратегии выбора параметров релевантности.

Определены принципы компьютерной реализации FDLP-структур. Проведены эксперименты, демонстрирующие эффективность теории.

## Список литературы

[1] *Махортов, С. Д.* Математические основы искусственного интеллекта: теория LP-структур для построения и исследования моделей знаний продукционного типа / Под ред. В. А. Васенина. — М. : МЦНМО, 2009. — 304 с.

[2] *Ferré, S.* Formal Concept Analysis: From Knowledge Discovery to Knowledge Processing / S. Ferré, M. Huchard, M. Kaytoue [et al.] // A Guided Tour of Artificial Intelligence Research / Eds. Marquis P., Papini O., Prade H. — Cham : Springer, 2020. — Vol. 2. — P. 411–435.

[3] *Halib M.* Bit-vector encoding for partially ordered sets / M. Halib, L. Nourine // Lecture Notes in Computer Science. — 1994. — Vol. 831. — P. 1–12.

[4] *Makhortov, S. D.* Algebraic Models for Big Data and Knowledge Management // 22nd International Conference on Data Analytics and Management in Data Intensive Domains (DAMDID/RCDL 2020), Selected Proceedings. CCIS. — Cham : Springer, 2021. — Vol. 1427. — P. 19–26.

**Библиографическая ссылка**

*Махортов, С. Д.* Основанные на решетках алгебраические системы и их приложения в задачах управления знаниями // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 50–56.
https://doi.org/10.26456/mfcsics-21-6

**Сведения об авторах**

Сергей Дмитриевич Махортов
Воронежский государственный университет. Заведующий кафедрой

*Россия, 394018, Воронеж, Университетская пл., 1, ВГУ*
*E-mail: msd_exp@outlook.com*

УДК 510.6, 510.82, 510.53
AMS MSC2020: 03B70, 68T27, 68Q17, 68T30, 03D15

# Декомпозиция логических теорий: вычислительные проблемы и приложения

**Пономарев Д. К.**

Институт систем информатики им. А. П. Ершова СО РАН;
Новосибирский государственный университет

Аннотация. Интерес к методам декомпозиции в логике связан с анализом и применением больших аксиоматических теорий, возникающих в приложениях. Применение логики многогранно и исследование свойств, связанных с декомпозицией, требует разнообразных техник. Эта тема стала актуальной, главным образом, благодаря развитию логических методов представления знаний и автоматизированного вывода, однако полученные результаты оказались интересными как для классической области логики, так и канонических приложений, например, синтеза логических схем и компрессии данных.

Ключевые слова: прикладная логика, дискретная математика, сложность вычислений, декомпозиция Булевых функций, аксиоматическое представление знаний, инженерия онтологий, компрессия данных.

Некоторые аксиоматические теории, известные в математике, представлены как объединение двух (или нескольких) теорий, имеющих дизъюнктные сигнатуры. Например, теория упорядоченных графов — есть объединение теории бинарного предиката, задающего отношение смежности, и теории полного порядка. Такое дизъюнктное представление довольно редко встречается для теорий в математике, поскольку аксиомы, как правило, формулируют связи между предикатами и операциями. Например, в определении упорядоченных алгебраических структур, таких как группы или кольца, обычно используется аксиома, постулирующая связь между порядком и операцией в виде: «$a$ меньше либо равно $b$ влечет $a + c$ меньше либо равно $b + c$».

По-иному выглядит ситуация в областях Computer Science и Knowledge Representation & Reasoning, где логические теории используются для описания сложных систем из разных (не только строго математических) предметных областей и они имеют более сложный вид и гораздо большее число аксиом, в сравнении с теориями «стандартных» алгебраических структур [3]. Кроме того, такие теории зачастую разрабатываются совместно группами экспертов предметной области или они формируются полностью автоматически, например, из текстов. Это относится, например, к логическим теориям — онтологиям, которые представляют формализацию характерных понятий и отношений, встречающихся в предметных областях, таких как биология, медицина и пр. Такие теории могут включать тысячи сигнатурных символов (имен понятий и отношений) и сотни тысяч аксиом, что делает анализ таких теорий и логический вывод из них довольно нетривиальными задачами [4, 6].

Возникает естественный вопрос: можно ли как-то структурировать теорию, имеющую большой набор аксиом, можно ли упростить работу с теорией, представив ее в более удобном виде. Например, известно, что арифметика Пеано — довольно выразительная и сложная теория с алгоритмической точки зрения из-за связей между операциями сложения и умножения. Однако арифметика Пресбургера (в которой нет операции умножения) существенно менее сложна. Теория двух унарных функций неразрешима, но теория одной унарной функции является разрешимой (на самом деле, даже монадическая теория второго порядка одной унарной функции разрешима).

Это наводит на мысль, что даже если теория сложна, то ее компоненты, получаемые как «проекции» на подмножества сигнатурных символов, могут оказаться гораздо проще. Как определить, представима ли теория в виде объединения такого рода компонент и как вычислить сами компоненты? Ответ на этот вопрос затрагивает классические и новейшие результаты из области логики, теории моделей, теории вычислений, дискретной математики и открывает путь к новым методам синтеза логических схем [1], компрессии данных [2], инженерии онтологий [5] и теорий действий [7].

**Список литературы**

[1] *Emelyanov, P.* The Complexity of AND-decomposition of Boolean Functions / P. Emelyanov, D. Ponomaryov // Discrete Applied Mathematics. — 2020. — Vol. 280. — P. 113–132.

[2] *Emelyanov, P.* On Two Kinds of Dataset Decomposition. // Computational Science — ICCS 2018. ICCS 2018. LNCS'10861 / Eds. Shi Y. [et al.] — Cham : Springer, 2018. — P. 171–183.

[3] Handbook of Knowledge Representation / Eds. F. van Harmelen, V. Lifschitz, B. Porter. — Amsterdam : Elsevier, 2008.

[4] Handbook on Ontologies / Eds. S. Staab. R. Studer. — Berlin : Springer, 2009.

[5] *Konev, B.* Decomposing Description Logic Ontologies / B. Konev, C. Lutz, D. Ponomaryov, F. Wolter // Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010. — AAAI Press, 2010.

[6] *Lutz, C.* Mathematical Logic for Life Science Ontologies / C. Lutz, F. Wolter // Logic, Language, Information and Computation. WoLLIC 2009. LNCS'5514 / Eds. Ono H., Kanazawa M., de Queiroz R. — Berlin : Springer, 2009.

[7] *Ponomaryov, D.* Progression of Decomposed Local-Effect Action Theories / D. Ponomaryov, M. Soutchanski // ACM Transactions on Computational Logic. — 2017. — Vol. 18, iss. 2. — 16.

**Библиографическая ссылка**

*Пономарев, Д. К.* Декомпозиция логических теорий: вычислительные проблемы и приложения // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 57–60.
https://doi.org/10.26456/mfcsics-21-7

**Сведения об авторах**

Денис Константинович Пономарев
Институт систем информатики им. А. П. Ершова СО РАН;
Новосибирский государственный университет. Старший научный сотрудник

*пр. Лаврентьева, 6, 630090, Новосибирск*
*E-mail: ponom@iis.nsk.su*

УДК 510.635
AMS MSC2020: 68Q45

# Теория определимости в контексте информационно-коммуникационных систем[1]

**Семенов А. Л.**

Московский государственный университет им. М. В. Ломоносова;
Институт кибернетики и образовательной информатики им. А. И. Берга ФИЦ ИУ РАН;
Московский физико-технический институт (национальный исследовательский университет);
Институт математики и механики им. Н. И. Лобачевского, Научно-образовательный математический центр Приволжского федерального округа

Аннотация. В работе обсуждается проблематика определимости и пространств отношений в исторической перспективе, обрисована роль Альфреда Тарского и Ларса Свенониуса, рассматриваются последние результаты, расширяющие полученные ранее для однородных структур, в частности на случай пополнимых вверх. Приложения включают языки описания баз данных, анализ CSP — Constraint Satisfaction Problem (обобщенной выполнимости).

Ключевые слова: теория определимости, Альфред Тарский, пространства определимости, редукты, полные вверх структуры, теорема Свенониуса, базы данных, реляционные алгебры, CSP.

## Введение

В докладе будет идти речь о последних результатах и, главное, об открытых проблемах теории определимости. Лейбниц мечтал о lingua generalis — универсальном языке, в котором можно определить любые понятия [1]. Частичным осуществлением этой мечта стала

[1]Работа выполнена при финансовой поддержке РНФ, грант № 17-11-01377.

конструкция Анны Вержбицкой универсальной системы базовых понятий любого языка [18, 19].

## 1. История вопроса

Конец XIX–начало XX века ознаменовались поиском базовой системы определений для математики наряду с поисками адекватной системы доказательства. Результатами в области определений стали:

- Канторовская математика, где все свелось к одному виду объектов (множеств) и одному двухместному отношению (принадлежности).
- Арифметика Пеано.
- Геометрия, где итальянской школой, Марио Пиери и др. были построены замечательные системы, за которыми следовало построение Гильберта, интерес Тарского привел, в частности, к прояснению вопроса о том, что двуместных отношений для построения геометрии недостаточно (результат Линденбаума–Тарского).

В 1900 г. в Париже последовательно прошли два конгресса — Международный конгресс по философии и Международный конгресс математиков (где Гильберт поставил свои Проблемы). На каждом из этих конгрессах определимость была темой нескольких докладов. В частности, Алессандро Падоа в своем докладе «Логическое введение в любую дедуктивную теорию» предложил метод доказательства неопределимости какого-то отношения через другие [14]. По существу, это был метод автоморфизмов, центральный в данной теории.

Эдвард Хангтинтон в свой статье 1916 г. [13] писал: «Существуют четыре типа порядка, важных в геометрии и других областях математики: линейный порядок, «между»; цикл; разделение двух пар точек».

Все это вместе заложило основу для теории определимости.

Тарский сохранил интерес к определимости на протяжении всей своей жизни, удивляясь тому, что, как он писал, «математики относятся к понятию определимости с недоверием и подозрительностью» [16].

Одним из вкладов Тарского и его ближайшего коллеги Адольфа Линдебаума в теорию определимости было построение алгебры Линденбаума – Тарского, «бескоординатного» варианта понятия логически определяемого отношения. Результат Тарского (и Геделя) о неопределимости в арифметике арифметической истины, как теорема Тарского об элиминации кванторов, конечно, относятся к жемчужинам теории определимости.

## 2. Теорема Свенониуса

Существует естественное антимонотонное соответствие Галуа между пространствами определимости и группами перестановок универсума — надгруппами группы автоморфизмов исходной структуры.

«Теоремой полноты» для определимости стала теорема Свенониуса, опубликованная в 1959 г. в журнале Theoria по философии науки, издававшемся университетом Лунда [15]. Теорема утверждает, что метод автоморфизмов всегда позволяет установить не определимость, если допустить элементарные расширения — добавление «идеальных», «мнимых» элементов. Специалисты, например, Р. Бюхи и К. Данхоф [7] оценили результат Свенониуса именно как теорему полноты для определимости, связывая этот результат с Эрлангенской программой Клейна и отмечая медленное признание математиками важности результата Свенониуса. Сегодня недооцененность этой теоремы видна хотя бы из того, что статьи Википедии, относящиеся к определимости, содержат аккуратные определения, в частности — автоморфизма, но даже не упоминают теорему Свенониуса. В Википедии все же есть посвященная Свенониусу статья, где говорится о «'Svenonius theorem' on decidability».

Тем не менее, определенный ренессанс в теории определимости, начиная с 1960-х годов возник. Он был связан, помимо теоремы Свенониуса, с теорией конечных автоматов и определяемых ими отношениями (хотя и в этой области ссылок на Свенониуса не найти). Многие задачи здесь формулировались фактически в терминах теории определимости.

## 3. Решетки определимости. Автоморфизмы

Начавшаяся цифровая эра сразу же привела к применению компьютеров к поиску информации. Поиск требовал упорядочения информации, например снабжения объектов атрибутами. Это быстро привело к реляционным (то есть — «отношенным») базам данных, иными словами — к алгебрам отношений (реляционные алгебры Кодда и т. д.), а также — к соответствующим логическим системам.

При рассмотрении вопросов об определимости/неопределимости одних отношений через другие, нам представляется разумным (следуя Тарскому) использовать бескоординатный подход. При таком подходе, начав с некоторого семейства отношений на каком-то универсуме, мы рассматриваем замыкание этого множества. Дав имена конечному подмножеству отношений, можно написать формулу (в логике первого порядка). Эта формула задает отношение на универсуме. Все получающиеся так отношения составляют пространство определимости. Естественно, возникает задача описания решеток подпространств определимости данного пространства (которое может быть задана как реляционная структура).

В 1965 г. Клод Фрасне получил первое описание решетки определимости [12]. Это была решетка определимости порядка рациональных чисел, порождающими элементами для пространств были отношения Хантингтона, к котором добавляется тривиальный минимальный элемент — равенство. В последующие годы были получены многочисленные результаты по решеткам определимости. Во всех этих результатах рассматривались однородные структуры, для таких структур имеется антиизоморфизм решетки определимости и решетки замкнутых (в естественной топологии) надгрупп. Все найденные решетки оказались конечными. Гипотеза Томаса (1991 г.) состоит в том, что конечность имеет место для всех решеток однородных структур [17].

Однородные структуры с конечной сигнатурой омега-категоричны: все структуры, им элементарно эквивалентные, изоморфны. Развивая проблематику дальше, мы предложили рассмотреть структуры, для которых изоморфны все их элементарные расширения. Такие структуры мы называем *полными вверх*. Если структура имеет полное вверх элементарное расширение, то ее решетка

определимости антиизоморфна структуре надгрупп автоморфизмов для этого расширения.

Рассмотрение полных вверх структур позволило нам впервые получить описания решеток определимости для неоднородных структур, например, для следования целых чисел. Эти решетки оказались бесконечными. Некоторое обобщение понятия однородности позволяет строить соответствующие пополнения.

## 4. Специальные ситуации, ограничения и приложения

Стараясь найти реалистичные ограничения исходных постановок, можно рассматривать структуры с ограниченной логической сложностью. Так например, для омега-категоричных структур имеет место элиминация кванторов. Однако требование элиминируемости кванторов в случае конечной реляционной сигнатуры оказывается слишком ограничительным (например, ему не удовлетворяет арифметика сложения целых чисел). В работах автора, относящихся к структурам на натуральном ряде в качестве естественного обобщения рассматривались структуры с конечной (в частности — реляционной) сигнатурой, где всякая формула эквивалентна экзистенциальной [2–4]. Другой подход, использованный, в частности, в автоматных моделях баз данных был предложен в работе [10], где предлагается ограничивать число аргументов символов отношений и функций, но допустить возможность их бесконечного количества при элиминации кванторов. Представляет интерес сравнение двух указанных подходов.

Еще одним подходом, приближающим нас к практическим задачам, является рассмотрения свойства «обобщенной выполнимости» или «задачи удовлетворения ограничениям» — Constraint Satisfaction Problem (CSP) — выяснения вложимости конечной структуры в заданную. CSP эквивалентна определимости с помощью ограниченного класса формул: экзистенциальных от позитивных конъюнкций — ограничений. И здесь также большой объем исследований был выполнен для случая однородных структур и пространств определимости, в частности, была сформулирована гипотеза о дихотомии в сложности решения CSP [11]. Проблема была решена независимо А. Булатовым [8, 9] и Д. Жуком [20] для вложимости в конечную

структуру. В случае вложений в однородные структуры она была поставлена М. Бодирским, М. Пинскером и А. Понграцем [5]. В работе [6] она формулируется, как Infinite domain CSP dichotomy conjecture для специального класса пространств в однородных структурах. По-видимому, она имеет место и для некоторых структур, пополняемых вверх.

**Список литературы**

[1] *Лейбниц, Г. В.* О словах. Пер.с фр. — М. : Книжный дом «ЛИБРКОМ», 2010. — 96 с.

[2] *Семенов, А. Л.* О некоторых расширениях арифметики сложения натуральных чисел // Известия Академии наук СССР. Серия математическая. — 1979. — Т. 43, № 5. — С. 1175–1195.

[3] *Семенов, А. Л.* Логические теории одноместных функций на натуральном ряде // Известия Академии наук СССР. Серия математическая. — 1983. — Т. 47, вып. 3. — С. 623–658.

[4] *Семенов, А. Л.* Решетка определимости (редуктов) для целых чисел с операцией следования / А. Л. Семенов, С. Ф. Сопрунов // Известия РАН. Серия математическая. — 2021. — № 85:6. (в печати)

[5] *Bodirsky, M.* Projective clone homomorphisms / M. Bodirsky, M. Pinsker, A. Pongrácz // Journal of Symbolic Logic. — 2021. — Vol. 86. — № 1. — С. 148–161.

[6] *Bodor, B.* Classification of $\omega$-categorical Monadically Stable Structures. — 2020. — URL: arXiv:2011.08793v1. — Загл. с титул. экрана.

[7] *Büchi, J. R.* Definibility in Normal Theories / J. R. Buchi, K. J. Danhof // Israel Journal of Mathematics. — 1973. — № 14.3. — P. 248–256.

[8] *Bulatov, A. A.* A Dichotomy Theorem for Nonuniform CSPs // In 58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017. — 2017.

[9] *Bulatov, A. A.* A Dichotomy Theorem for Nonuniform CSPs Simplified. — URL: arXiv:2007.09099. — Загл. с титул. экрана.

[10] Definable Relations and First-order Query Languages Over Strings / M. Benedikt, L. Libkin, T. Schwentick, L. Ségoufin // Journal of the ACM. — 2003. — Vol. 50. — P. 694–751.

[11] *Feder, T.* The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study Through Datalog and Group Theory / T. Feder, M. Y. Vardi // SIAM Journal of Computing. — 1998. — V. 28. — P. 57–104.

[12] *Frasnay, C.* Quelques Problèmes Combinatoires Concernant les Ordres Totaux et les Relations Monomorphes // Annales de l'institut Fourier. — 1965. — Vol. 15, № 2. — P. 415–524.

[13] *Huntington, E. V.* Inter-Relations Among the Four Principal Types of Order // Transactions of the American Mathematical Society. — 1935. — T. 38.1. — P. 1–9.

[14] *Padoa, A.* Logical Introduction to Any Deductive Theory // 1900. Опубл. в сб. From Frege to Gödel. A Source Book in Mathematical Logic, 1879–1931. Ed. by Jean van Heijenoort. Cambridge, Mass., Harvard University Press, 1967. — P. 118–123.

[15] *Svenonius, L.* A Theorem on Permutations in Models // Theoria. — 1959. — T. 25.3. — P. 173–178.

[16] *Tarski, A.* Sur les Ensembles Définissables de Nombres Réels // Fundamenta Mathematicae. — 1931. — Vol. 17, № 1. — C. 210–239.

[17] *Thomas, S.* Reducts of the Random Graph // Journal of Symbolic Logic. — 1991. — Vol. 56(1). — P. 176–181.

[18] *Wierzbicka, A.* What Did Jesus Mean?: Explaining the Sermon on the Mount and the Parables in Simple and Universal Human Concepts. — New York : Oxford University Press, 2001. — 512 с.

[19] *Wierzbicka, A.* Imprisoned in English: The Hazards of English as a Default Language. — New York : Oxford University Press, 2013. — 304 с.

[20] *Zhuk, D.* A Proof of the CSP Dichotomy Conjecture // Journal of the ACM. — 2020. — Vol. 67, № 5. — C. 1–78.

**Библиографическая ссылка**

*Семенов, А. Л.* Теория определимости в контексте информационно-коммуникационных систем // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 61–68.
https://doi.org/10.26456/mfcsics-21-8

**Сведения об авторах**

Алексей Львович Семенов
Московский государственный университет им. М. В. Ломоносова;
Институт кибернетики и образовательной информатики им. А. И. Берга ФИЦ ИУ РАН;
Московский физико-технический институт (национальный исследовательский университет);
Институт математики и механики им. Н. И. Лобачевского, Научно-образовательный математический центр Приволжского федерального округа. Зав. кафедрой математической логики и теории алгоритмов МГУ им. М. В. Ломоносова

*Ленинские горы, 1, Москва 119991, Россия*
*E-mail: alsemno@ya.ru*

УДК 510.52, 510.662
AMS MSC2020: 03F20, 03D15

# Несколько слов о сложности доказательств

**Соколов Д. О.**

Санкт-Петербургский государственный университет;
Санкт-Петербургское отделение Математического института им. В. А. Стеклова РАН

Аннотация. Теория сложности доказательств изучает длины доказательств пропозициональных формул. За последние тридцать лет в данной области произошло ряд существенных прорывов, а также были открыты новые связи с другими разделами теории сложности вычислений. Мы рассмотрим, как основные задачи теории сложности доказательств, так и примеры применений.

Ключевые слова: теория сложности, системы доказательств, метод резолюций, булевы схемы.

Один из наиболее естественных вопросов математической логики: по заданному истинному утверждению оценить длину кратчайшего доказательства в какой-нибудь аксиоматической системе. Мы сосредоточимся на этом вопросе для пропозициональной логики, основном объекте теории сложности доказательств. Для удобства вместо истинных утверждений (тавтологий) мы перейдем ко всюду ложным утверждениям и будет рассматривать «доказательства» невыполнимости пропозициональных формул. Следуя терминологии теории сложности, мы будем изучать язык $\mathsf{UNSAT}$ невыполнимых пропозициональных формул в КНФ.

Начнем с основного определения, которое было сформулировано в работе [3].

Определение 1. *Системой доказательств для языка* $\mathsf{UNSAT}$ *будем называть такую полиномиально вычислимую функцию* $\Pi\colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$, *что:*

- *если* $\varphi \in \mathsf{UNSAT}$, *то существует такое* $w \in \{0,1\}^*$, *что* $\Pi(\varphi, w) = 1$ *(будем говорить, что* $w$ *— это доказательство для* $\varphi$*);*
- *если* $\varphi \notin \mathsf{UNSAT}$, *то для всех* $w \in \{0,1\}^*, \Pi(\varphi, w) = 0$.

Про системы доказательств можно думать в терминах игр. Рассмотрим двух игроков: Оптимист и Пессимист, которые получают формулу $\varphi$ в КНФ от $n$ переменных $x_1, \ldots, x_n$. Оптимист считает, что есть некоторый выполняющие набор для формулы $\varphi$, то есть такой набор значений $a_1, a_2, \ldots, a_n \in \{0, 1\}$, что $\varphi(a_1, a_2, \ldots, a_n) = 1$. А Пессимист считает что такого набора нет и пытается убедить в этом Оптимиста, предъявив некоторое доказательство $w$. При этом система доказательств определяется тем, какие доказательства Оптимист считает корректными.

Одним из классических примеров систем доказательств является резолюционная система. В данной системе доказательство $\pi$ для формулы $\varphi$ представляет собой такую упорядоченную последовательность дизъюнктов $\pi := D_1, \ldots, D_s$, что $D_s = \varnothing$ пустой дизъюнкт и для каждого $i \in [s]$ либо $D_i$ это дизъюнкт $\varphi$, либо найдутся такие $j, k < i$, что $D_i$ получен из $D_j$ и $D_k$ путем применения резолюционного правила

$$\frac{C \vee x \qquad D \vee \neg x}{C \vee D}$$

или правила ослабления

$$\frac{C}{D}\text{, где } [C \subseteq D].$$

В терминах нашего определения П — это алгоритм, который получает вместо формулу $\varphi$, и в качестве $w$ он получает резолюционное доказательство, которое ему необходимо проверить.

Как мы уже замечали в начале, основной задачей сложности доказательств является оценка размера кратчайших доказательств невыполнимости формул в различных системах. И сложность доказательств — один из немногих разделов теории сложности, где удается получить безусловные нижние оценки. В частности: на резолюционную систему доказательств [5], на ряд систем алгебраического вывода [2, 6, 9]. Однако, для некоторых систем вопрос о нижних оценках по прежнему открыт, например для системы Фреге (Гильбертовская система).

Нижние и верхние оценки на сложность доказательств в различных системах часто удается переносить на другие модели вычислений.

1) Если бы нам удалось привести пример формул для которых кратчайшее доказательство имеет суперполиномиальный размер (от длины формулы) во всех системах, то это бы повлекло за собой неравенство классов $\mathsf{NP} \neq \mathsf{coNP}$ [3], и, в частности, классов $\mathsf{P} \neq \mathsf{NP}$, что является одной из «задач тысячелетия».
2) Даже если мы сосредоточимся на резолюционной системе доказательств, то нижние оценки размер доказательств в ней влекут нижние оценки на время работы популярных алгоритмов для решения задачи выполнимости булевых формул (что является $\mathsf{NP}$-полной задачей) [1, 4].
3) Также, известных нижних оценок (на резолюционную систему, а также на ряд систем, основанных на алгебраическом выводе) хватает для получения сильных нижних оценок на монотонные модели вычисления, в частности на монотонные схемы и, так называемые, монотонные *span programs* [7, 8].

В докладе мы сосредоточимся на применения теории сложности, а также обсудим основные задачи и современные проблемы данной теории.

**Список литературы**

[1] *Alekhnovich, M.* Exponential Lower Bounds for the Running Time of DPLL Algorithms on Satisfiable Formulas / M. Alekhnovich, E. A. Hirsch, D. Itsykson // Journal of Automated Reasoning. — 2005. — Vol. 35. — P. 51–72.

[2] *Clegg, M.* Using the Groebner basis algorithm to find proofs of unsatisfiability / M. Clegg, J. Edmonds, R. Impagliazzo // Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing (STOC '96). — New York : Association for Computing Machinery, 1996. — P. 174–183.

[3] *Cook, S. A.* The Relative Efficiency of Propositional Proof Systems / S. A. Cook, R. A. Reckhow // The Journal of Symbolic Logic. — 1979. — Vol. 44, iss. 1. — P. 36–50.

[4] *Davis, M.* A Computing Procedure for Quantification Theory / M. Davis, H. Putnam // Journal of the ACM. — 1960. — Vol. 7, iss. 3. — P. 201–215.

[5] *Haken, A.* The intractability of resolution // Theoretical Computer Science. — 1985. — Vol. 39. — P. 297–308.

[6] *Impagliazzo, R.* Lower Bounds for the Polynomial Calculus and the Gröbner Basis Algorithm / R. Impagliazzo, P. Pudlák, J. Sgall // Computational Complexity. — 1999. — Vol. 8. — P. 127–144.

[7] Monotone circuit lower bounds from resolution / A. Garg, M. Göös, P. Kamath, D. Sokolov // Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2018) / Eds. Ilias Diakonikolas, David Kempe, Monika Henzinger. — New York : Association for Computing Machinery, 2018. — P. 902–911.

[8] *Pitassi, T.* Lifting nullstellensatz to monotone span programs over any field. / T. Pitassi, R. Robere // Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2018) / Eds. Ilias Diakonikolas, David Kempe, Monika Henzinger. — New York : Association for Computing Machinery, 2018. — P. 1207–1219.

[9] Proof complexity in algebraic systems and bounded depth Frege systems with modular counting / S. Buss, R. Impagliazzo, J. Krajíček [et al.] // Computational Complexity. — 1997. — Vol. 6, iss. 3. — P. 256–298.

### Библиографическая ссылка

*Соколов, Д. О.* Несколько слов о сложности доказательств // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 69–72.
https://doi.org/10.26456/mfcsics-21-9

### Сведения об авторах

Дмитрий Олегович Соколов
Санкт-Петербургский государственный университет;
Санкт-Петербургское отделение Математического института им. В. А. Стеклова РАН. Доцент; научный сотрудник

*Санкт-Петербург, 14-я линия Васильевского острова 29*
*E-mail: sokolov.dmt@gmail.com*

УДК 510.5
AMS MSC2020: 03D30

# Тотальные и кототальные степени перечислимости

**Солон Б. Я.**

Ивановский государственный университет

Аннотация. Для произвольного множества $A \subseteq \omega$ e-степень множества $A$ и e-степень его дополнения $\bar{A}$ не обязаны быть сравнимы. Чтобы обеспечить сравнимость, можно выделить два класса e-степеней. Первый был введен одновременно с самой сводимостью по перечислимости — это класс тотальных e-степеней. По одному из определений множество $A \subseteq \omega$ тотально, если $\bar{A} \leqslant_e A$, и e-степень тотальна, если она содержит некоторое тотальной множество. Второй класс связан с перестановкой множеств $A$ и $\bar{A}$ в этом отношении e-сводимости. Первыми выделили множества $A$ с свойством $A \leqslant_e \bar{A}$ Ж. Миллер и М. Соскова в 2010 г. Для этих множеств они ввели термин кототальные множества. e-степень называется кототальной, если она содержит некоторое кототальное множество. В докладе будут расмотрены некоторые свойства обоих классов e-степеней.

Ключевые слова: сводимость по перечислимости, e-степень, тотальное множество, кототальное множество.

## Введение

Мы будем использовать понятия и терминологию, которые приняты в монографии [7]. В статье [5] авторы впервые всесторонне рассмотрели понятия тотальности множеств и e-степеней, которые появились вполне естественно вместе с понятием e-сводимости, и понятие кототальности, которое впервые использовалось (как термин) в тезисах А. В. Панкратова [6] (в то время — моего аспиранта) и было изучено более широко в статьях автора [1, 8]. В статье [5] для множеств была введена система терминов, характеризующих различные уровни «кототальности» множеств — это графкототальность, кототальность и слабая кототаль-

ность. В связи с предложенной новой терминологией появилась необходимость пересмотреть результаты работ [1, 6, 8]

## 1. Основные определения

Кроме определений, приведенных в аннотации, дадим дополнительно те, которые будут использованы в докладе. Пусть $\omega = \{0, 1, 2, \dots\}$ — множество натуральных чисел, $A \subseteq \omega$. Функция $f : \omega \to \omega$ называется тотальной, если $\mathrm{dom}(f) = \omega$. Обозначим через $TF$ множество всех тотальных функций.

Определение 1. *Множество $A$ называется граф-кототальным, если $A = \overline{\mathrm{graph}(f)}$ для некоторой функции $f \in TF$.*

Определение 2. *Множество $A$ называется слабо кототальным, если $\bar{A} \equiv_e \mathrm{graph}(f)$ для некоторой функции $f \in TF$.*

Определение 3. *e-степень $\mathbf{a}$ называется граф-кототальной (слабо кототальной), если она содержит некоторое граф-кототальное (слабо кототальное) множество.*

Определение 4. *e-степень $\mathbf{a}$ называется квазиминимальной, если она ненулевая и единственная тотальная e-степень ниже $\mathbf{a}$ равна $\mathbf{0} = \deg_e(\varnothing)$.*

## 2. Основные результаты

Ясно, что каждая кототальная e-степень является слабо кототальной, а граф-кототальная e-степень — кототальной. В [5] показано, что все эти три уровня кототальности являются различными.

Класс граф-кототальных e-степеней лежит строго между тотальными степенями и кототальными степенями. Чтобы увидеть, что каждая тотальная степень является граф-кототальной, достаточно заметить, что каждая тотальная степень содержит график характеристической функции $c_A$ некоторого тотального множества $A$; она также содержит дополнение графика $c_A$.

Теорема 1. *Любая тотальная e-степень $\mathbf{a} \geqslant \mathbf{0}'$ содержит функцию $f \in TF$ такую, что $\deg_e(\overline{\mathrm{graph}(f)})$ — квазиминиимальная e-степень.*

Теорема 2. *Для каждой тотальной e-степени $\mathbf{b} \geqslant \mathbf{0}'$ существует граф-кототальная квазиминимальная e-степень $\mathbf{a}$ такая, что $\mathbf{a}' = \mathbf{b}$.*

Замечание 1. *Эта теорема усиливает результат К. Макэвоя [4], который доказал, что квазиминимальные e-степени имеют все возможные e-скачки.*

Теорема 3. *Для каждой тотальной e-степени* $\mathbf{b}$ *существует граф-кототальная квазиминимальная e-степень* $\mathbf{a}$ *над* $\mathbf{b}$.

Замечание 2. *Эта теорема усиливает результат Л. Гаттериджа [2] о существовании квазиминимальных e-степеней.*

## Заключение

В последнее время было опубликовано большое количество результатов, содержащих примеры ко-тотальных множеств и e-степеней. Отмечу результат Жанделя [3] о том, что множество неидентичных слов в конечно порожденной простой группе кототально. Данное направление в изучении e-степеней можно продолжить, рассматривая различные уровни «тотальности» и «кототальности» функций и их частичных степеней.

## Список литературы

[1] *Солон, Б. Я.* Тотальные и ко-тотальные степени перечислимости // Известия высших учебных заведений. Математика. — 2005. — № 9. — С. 60–68.

[2] *Gutteridge L.* Some results on enumeration reducibility : Ph. D. Dissertation. — Vancouver : Simon Fraser University, 1971.

[3] *Jeandel E.* Enumeration reducibility in closure spaces with applications to logic and algebra // Proc. 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS). — 2017. — P. 1–11.

[4] *McEvoy K.* Jumps of quasiminimal enumeration degrees // Journal of Symbolic Logic. — 1985. — Vol. 50, № 3. — P. 839–848.

[5] On cototality and the skip operator in the enumeration degrees / U. Andrews, H. A. Ganchev, R. Kuyper [et al.] // Transactions of the American Mathematical Society. — 2019. — Vol. 372. — P. 1631–1670.

[6] *Pancratov, A. V.* Some properties of e-degrees of cototal sets // Int. conf. «Logic and applications», Proceedings. — Novosibirsk, 2000.

[7] *Rogers, H. Jr.* Theory of Recursive Functions and Effective Computability. — New York : McGraw-Hill, 1967. — 482 p.

[8] *Solon, B.* Co-total Enumeration Degrees // Logical Approaches to Computational Barriers. CiE 2006. LNCS'3988. / Eds. Beckmann A. [et al.] — Berlin : Springer, 2006. — P. 538–545.

**Библиографическая ссылка**

*Солон, Б. Я.* Тотальные и кототальные степени перечислимости // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 73–76.
https://doi.org/10.26456/mfcsics-21-10

**Сведения об авторах**

Борис Яковлевич Солон
Ивановский государственный университет. Заведующий кафедрой

*Россия, 153025, г. Иваново, ул. Ермака, 39, ЦФО*
*E-mail: bysolon@gmail.com*

УДК 510.67
AMS MSC2020: 03C30, 03C15, 03C50

# Семейства элементарных теорий и их характеристики[1]

**Судоплатов С. В.**

Институт математики им. С. Л. Соболева СО РАН;
Новосибирский государственный технический университет;
Новосибирский государственный университет

Аннотация. Дается обзор результатов о семействах элементарных теорий, их комбинациях и основных характеристиках, включая ранги и степени как в абсолютном смысле, так и относительно заданных формул. Исследуются связи между формулами и семействами теорий, а также связи между свойствами теорий при предельных переходах. Приводится механизм построения $n$-арных теорий через их аппроксимации.

Ключевые слова: элементарная теория, семейство теорий, формула, ранг, степень, $n$-арная теория.

## Введение

Теория моделей [2, 4, 5] сформировалась в 1930–1950-х годах как раздел математической логики [3, 6], в котором изучаются связи между формальным языком, задающим информацию в виде синтаксических объектов, основанных на множествах формул, и его интерпретациями, моделями или структурами, являющимися семантическими объектами. Эти объекты можно использовать для классификации друг друга, создавая структурные классификации теорий и их моделей [7, 19]. При решении вопросов классификации возникают значимые характеристики (размерности, ранги, спектры,

[1]Работа выполнена при частичной финансовой поддержке в рамках программы фундаментальных научных исследований СО РАН № I.1.1, проект № 0314-2019-0002, и Комитета науки Министерства образования и науки Республики Казахстан, грант № AP08855544.

различные меры сложности, и т.д.) для различных классов структур и их теорий [5, 20]. Универсальность предмета теории моделей позволяет, помимо собственных классификационных результатов, получать структурные результаты в смежных и прикладных областях.

Беря семейство структур (семантических объектов) или теорий (синтаксических объектов), можно определять как их влияние друг на друга через их комбинации [21, 22], так и возможности для построения новых структур/теорий по отношению к естественным операторам [23]. Эти операторы допускают аппроксимацию информации, рассматриваемой на семантическом/синтаксическом уровнях. В этом случае мы можем формировать структуры/теории с требуемыми свойствами, используя заданные аппроксимации и их предельные переходы [7, 24, 25].

Процесс построения новых структур/теорий может варьироваться с точностью до изоморфизма или элементарной эквивалентности. Количество этих вариаций определяет спектры/$e$-спектры для структур/теорий, которые могут иметь достаточно широкие диапазоны в допустимых рамках [5, 21]. Таким образом, с одной стороны, эти операторы могут генерировать новые структуры/теории, а с другой стороны, теоретико-модельные свойства и спектры/$e$-спектры для этих новых структур/теорий имеют границы, зависящие от заданных семейств.

К одной из важнейших характеристик теории $T$ относится значение ее арности $\mathrm{ar}(T)$ [27]. Эта характеристика связана с понятием базируемости теории [15] и исследована для различных теорий унаров и графов [8, 9], для упорядоченных теорий [1, 10].

Исследование семейств элементарных теорий и их характеристик включает следующие основные разделы, по которым получены определенные результаты:

1. Топологическая, спектральная и синтаксическая характеризация тотальной трансцендентности семейств теорий и их замыканий, как в целом для полных и неполных теорий, так и для семейств теорий абелевых групп [11, 12, 16, 18, 25].

2. Описание ранговых значений и их динамики для различных семейств теорий и их подсемейств [13, 14, 16, 25].

3. Описание аппроксимируемости и аппроксимаций теорий различными семействами [17, 24].

4. Характеризация и описание порождающих множеств, $P$-замыканий $\mathrm{Cl}_P(\mathcal{T})$ и $E$-замыканий $\mathrm{Cl}_E(\mathcal{T})$ для семейств теорий $\mathcal{T}$ [12, 23].

5. Характеризация и описание формул для семейств теорий, а также их характеристик [18, 26].

6. Описание арности теорий, их динамики и характеристик при переходе к замыканиям [27].

## Список литературы


[1] *Алтаева А. Б.* Бинарность почти $\omega$-категоричных вполне о-минимальных теорий / А. Б. Алтаева, Б. Ш. Кулпешов // Сибирский математический журнал. — 2020. — Т. 61, № 3. — С. 484–498.

[2] *Дудаков, С. М.* Основы теории моделей / С. М. Дудаков. — Тверь : Издательство Тверского государственного университета, 2013. — 480 с.

[3] *Ершов, Ю. Л.* Математическая логика / Ю. Л. Ершов, Е. А. Палютин. — М. : Физматлит, 2011. — 356 с.

[4] *Кейслер, Г.* Теория моделей / Г. Кейслер, Ч. Ч. Чэн. — М. : Мир, 1977. — 616 с.

[5] Справочная книга по математической логике. Ч. 1. Теория моделей / Под ред. Дж. Барвайса. — М. : Наука, 1982. — 392 с.

[6] *Судоплатов, С. В.* Математическая логика и теория алгоритмов : учебник и практикум для вузов / С. В. Судоплатов, Е. В. Овчинникова. — Изд. 5-е изд., стер. — М. : Изд-во Юрайт, 2021. — 256 с.

[7] *Судоплатов, С. В.* Классификация счетных моделей полных теорий. Ч. 1, 2 / С. В. Судоплатов. — Новосибирск : Изд-во НГТУ, 2018. — 376 + 452 с.

[8] *Судоплатов, С. В.* Базируемость стабильных теорий и свойства счетных моделей с мощными типами : дис... канд. физ.-мат. наук : 01.01.06 / С. В. Судоплатов. — Новосибирск, 1990. — 142 с.

[9] *Судоплатов, С. В.* Об одной оценке сложности теорий графов / С. В. Судоплатов // Сибирский математический журнал. — 1996. — Т. 37, № 3. — С. 700–703.

[10] *Kulpeshov B. Sh.* Criterion for binarity of $\aleph_0$-categorical weakly o-minimal theories / B. Sh. Kulpeshov // Annals of Pure and Applied Logic. — 2007. — Vol. 45. — P. 354–367.

[11] *Markhabatov, N. D.* Topologies, ranks, and closures for families of theories. I / N. D. Markhabatov, S. V. Sudoplatov // Algebra and Logic. — 2021. — Vol. 59, No. 6. — P. 437–455.

[12] *Markhabatov, N. D.* Topologies, ranks, and closures for families of theories. II / N. D. Markhabatov, S. V. Sudoplatov // Algebra and Logic. — 2021. — Vol. 60, No. 1. — P. 38–52.

[13] *Markhabatov, N. D.* Ranks for families of all theories of given languages / N. D. Markhabatov, S. V. Sudoplatov // Eurasian Mathematical Journal. — 2021. — Vol. 12, No. 2. — P. 52–58.

[14] *Markhabatov, N. D.* Definable subfamilies of theories, related calculi and ranks / N. D. Markhabatov, S. V. Sudoplatov // Siberian Electronic Mathematical Reports. — 2020. — Vol. 17. — P. 700–714.

[15] *Palyutin, E. A.* Models of superstable Horn theories / E. A. Palyutin, J. Saffe, S. S. Starchenko // Algebra and Logic. — 1985. — Vol. 24, No. 3. — P. 171–210.

[16] *Pavlyuk, In. I.* Ranks for families of theories of abelian groups / In. I. Pavlyuk, S. V. Sudoplatov // Bulletin of Irkutsk State University. Series Mathematics. — 2019. — Vol. 28. — P. 95–112.

[17] *Pavlyuk, In. I.* Approximations for theories of abelian groups / In. I. Pavlyuk, S. V. Sudoplatov // Mathematics and Statistics. — 2020. — Vol. 8, No. 2. — P. 220–224.

[18] *Pavlyuk, In. I.* Formulas and properties for families of theories of Abelian groups / In. I. Pavlyuk, S. V. Sudoplatov // Bulletin of Irkutsk State University. Series Mathematics. — 2021. — Vol. 36. — P. 95–109.

[19] *Shelah, S.* Classification theory and the number of non-isomorphic models / S. Shelah. — Amsterdam : North-Holland, 1990. — 705 p.

[20] *Sudoplatov, S. V.* Spectra for generative classes / S. V. Sudoplatov // Siberian Advances in Mathematics. — 2021. — Vol. 31, No. 1. — P. 53–68.

[21] *Sudoplatov, S. V.* Combinations of structures / S. V. Sudoplatov // Bulletin of Irkutsk State University. Series Mathematics. — 2018. — Vol. 24. — P. 82–101.

[22] *Sudoplatov, S. V.* Combinations of structures and of their theories (an informative survey) / S. V. Sudoplatov // Algebra and model theory 12. Collection of papers. — Novosibirsk : Edition of NSTU, 2019. — P. 86–127.

[23] *Sudoplatov, S. V.* Closures and generating sets related to combinations of structures / S. V. Sudoplatov // Bulletin of Irkutsk State University. Series Mathematics. — 2016. — Vol. 16. — P. 131–144.

[24] *Sudoplatov, S. V.* Approximations of theories / S. V. Sudoplatov // Siberian Electronic Mathematical Reports. — 2020. — Vol. 17. — P. 715–725.

[25] *Sudoplatov, S. V.* Ranks for families of theories and their spectra / S. V. Sudoplatov // Lobachevskii Journal of Mathematics. — 2021. — Vol. 42, No. 12. — P. 2959–2968.

[26] *Sudoplatov, S. V.* Formulas and properties, their links and characteristics / S. V. Sudoplatov // Mathematics. — 2021. — Vol. 9, Issue 12. 1391. — 16 pp.

[27] *Sudoplatov, S. V.* Arities and arizabilities of first-order theories / S. V. Sudoplatov. — Preprint, Novosibirsk, 2021.

### Библиографическая ссылка

*Судоплатов, С. В.* Семейства элементарных теорий и их характеристики // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 77–82.
https://doi.org/10.26456/mfcsics-21-11

### Сведения об авторах

Сергей Владимирович Судоплатов
Институт математики им. С. Л. Соболева СО РАН;

Новосибирский государственный технический университет;
Новосибирский государственный университет. Ведущий научный сотрудник; Заведующий кафедрой; Профессор

*Россия, 630090, Новосибирск, пр. Академика Коптюга, 4, ИМ СО РАН*
*E-mail: sudoplat@math.nsc.ru*

УДК 519.712
AMS MSC2020: 68W32

# Алгоритмы на строках и их связь с абстрактной алгеброй

**Тискин А. В.**

Санкт-Петербургский государственный университет

Аннотация. Рассматриваются алгоритмическая задачи сравнения строк и приближенного поиска в строке и их взаимосвязь с тропической матричной алгеброй и нестандартной разновидностью классической группы кос. Использование этой взаимосвязи позволяет получить эффективные алгоритмы приближенного поиска для сжатых строк, динамически изменяющихся строк, а также эффективные параллельные алгоритмы для данных задач.

Ключевые слова: алгоритмы на строках, сравнение строк, приближенный поиск в строке, унимонжевы матрицы, моноид Гекке.

В математике, неотъемлемой частью которой является теоретическая информатика, часто возникают интересные и удивительные связи между, казалось бы, непохожими и на первый взгляд не имеющими отношения друг к другу понятиями. Мы обсудим три таких понятия из трех разных областей:

- сравнение строк, приближенный поиск в строке по образцу — одни из наиболее известных и часто изучаемых алгоритмических задач, широко применяющиеся в биоинформатике;
- тропическая матричная алгебра — математика расстояний на плоскости с неожиданными правилами, где роль умножения играет сложение, а роль сложения — взятие минимума;
- абстрактные алгебраические структуры, задаваемые образующими и соотношениями — в данном случае речь будет идти об одной нестандартной версии классической группы кос.

Мы дадим соответствующие определения и увидим, как они оказываются тесно переплетены и взаимосвязаны между собой и

другими понятиями (геометрический поиск в прямоугольной области, комбинаторные сети сравнения), что позволяет лучше понимать и эффективнее решать многие теоретические и практические задачи — в частности, приближенный поиск в сжатых или динамически изменяющихся данных, локальное сравнение строк, вычисления на строках при помощи многопроцессорного и внутрипроцессорного параллелизма.

### Список литературы

[1] *Tiskin, A.* Fast Distance Multiplication of Unit-Monge Matrices // Algorithmica. — 2015. — Vol. 71. — P. 859–888.

[2] *Tiskin, A.* Communication vs Synchronisation in Parallel String Comparison // Proceedings of the 32nd ACM Symposium on Parallelism in Algorithms and Architectures. — 2020. — P. 479–489.

### Библиографическая ссылка

*Тискин, А. В.* Алгоритмы на строках и их связь с абстрактной алгеброй // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 83–84.
https://doi.org/10.26456/mfcsics-21-12

### Сведения об авторах

Александр Владимирович Тискин
Санкт-Петербургский государственный университет. Доцент

*Россия, 199034, Санкт-Петербург, Университетская наб., 7/9*
*E-mail: a.tiskin@spbu.ru*

УДК 519.681, 519.683, 519.712
AMS MSC2020: 03B70, 68N15

# Устранение рекурсии в полуинтерпретированных схемах программ

**Шилов Н. В.**

Университет Иннополис

Аннотация. В докладе представлен обзор результатов, полученных автором с начиная с 2010 г., по эффективному пребразванию («трансляции») паттернов рекурсивных программ (рекурсивных схем с неинтерпретированными или только частично интерпретированными функциональными и предикатными символами) в функционально эквивалентные стандартные схемы программ, то есть блок-схемы итеративных программ с теми же самыми неинтерпретированными или только частично интерпретированными функциональными и предикатными символами.

Ключевые слова: примитивно рекурсивные функции, рекурсивные функции, стандартные схемы программ, обогащенные схемы программ, рекурсивные схемы, функциональная эквивалентность схем программ.

## Введение

Примитивно рекурсивные функции — это минимальный класс функций на натуральных числах, который получается[1] из функции-константы 0, одноместной функции следования +1 и проекции (выбора элемента кортежа) при помощи операторов суперпозиции (композиции функций) и примитивной рекурсии

$$h(x_1,\dots x_n,y) \;=\; if\ y=0\ then\ f(x_1,\dots x_n) \\ else\ g\Big(x_1,\dots x_n,h\big(x_1,\dots x_n,(y-1)\big)\Big).$$

[1]Внимание: стандартная нотация для базисных функций и операторов не соблюдена!

Класс частично рекурсивных функций определяется аналогично классу примитивно рекурсивных, только к двум операторам (суперпозиции и примитивной рекурсии) добавляется еще оператор минимизации аргумента: $h(x_1, \dots x_n) = \arg\min\ y :\ f(x_1, \dots x_n, y) = 0$.

Будем говорить, что один (синтаксически определенный) класс функций[2] транслируем в некоторый другой (синтаксически определенный) класс функций[3], если любая функция из первого класса (функционально) эквивалентна некоторой функции из второго класса[4].

Разумеется, задача распознания по описанию частично рекурсивной функции ее «транслируемость» в класс примитивно рекурсивных функций является неразрешимой. Однако, исследование синтетически определенных программных паттернов, которые часто возникают при задании частично рекурсивных функций, и которые транслируются в итеративные программы (соответствующие примитивно рекурсивным функциям) вызывала [4] и по-прежнему вызывает интерес [3] в теории программирования и практике оптимизирующих компиляторов [2].

## 1. Рекурсивное динамическоое программирование

Динамическое программирование было введено Ричардом Беллманом в 1950-х годах для решения задач оптимального планирования. Уравнение Беллмана — это название рекурсивного функционального уравнения для целевой функции, которое выражает оптимальное решение в «текущем» состоянии через оптимальные решения в «достижимых за один шаг» состояниях, оно формализует так называемое Принцип оптимальности Беллмана: оптимальная программа (или план) остается оптимальной на каждом этапе. Мы изучаем класс уравнений Беллмана, который соответству-

[2]Например, класс частично рекурсивных функций.

[3]Например, класс примитивно рекурсивных функций.

[4]Здесь эквивалентность означает, что обе синтаксически определенные функции вычисляют одну и ту же функцию.

ет следующему рекурсивному шаблону:

$$G(x) \;=\; if\ p(x)\ then\ f(x) \\ else\ g\Big(x,\ \big\{h_i\big(x, G(t_i(x))\big),\ i \in [1..n(x)]\big\}\Big) \qquad (1)$$

Мы рассматриваем этот шаблон как рекурсивную программную схему [1], то есть рекурсивную структура управления с неинтерпретируемыми символами:

- $G$ — определяемый функциональный символ, представляющий (после интерпретации базисных функциональных и предикатных символов) целевую функцию $G : X \to Y$ для подходящих множеств $X$ и $Y$;

- $p$ — базисный предикатный символ, представляющий (после интерпретации) некоторый известный[5] предикат $p \subseteq X$;

- $f$ — базисный функциональный символ, представляющий (после интерпретации) некоторую известную[5] функцию (операцию) $f : X \to Y$;

- $g$ — базисный функциональный символ, представляющий (после интерпретации) некоторую известную[5] функцию (операцию) $g : X \times Z^* \to X$ для подходящего множества $Z$ (но переменной местности[6] $n(x) : X \to \mathbb{N}$);

- все $h_i$ и $t_i$ $(i \in [1..n(x)])$ — базисные функциональные символы, представляющие (после интерпретации) некоторые известные[5] функции $h_i : X \times Y \to Z$, $t_i : X \to X$ $(i \in [1..n(x)])$.

В дальнейшем мы не будем делать явного различия в обозначениях для символов и интерпретируемых символов, а будем писать/говорить, например, символ $g$ и/или функция $g$.

[5] То есть который/которую/которые уже умеем вычислять.
[6] То есть, фактически, от списка аргументов.

## 2. Основной результат

Теорема 1. *Предположим, что интерпретация базисных предикатных и функциональных символов в рекурсивной схеме (1) удовлетворяет следующим свойствам:*

- *количество аргументов* $n : X \to \mathbb{N}$ *— это некоторая константа* $n \in \mathbb{N}$*;*
- *все функции* $t_1, \ldots t_n$ *имеют обратные и* $t_i = (t_1)^i$ *для всех* $i \in [1..n]$*;*
- *предикат* $p$ *является* $t_1$*-замкнутым, то есть из* $p(u)$ *следует* $p(t_1(u))$ *для всех* $u \in X$*.*

*Пусть* $m \in \mathbb{N}$ *— число переменных достаточное для вычисления всех базисных предиката и функций* $p$, $f$, $h_i$ $(i \in [1..n])$, $t_1$ *и* $t_1^-$. *Тогда целевая функция* $G$ *может быть вычислена в этой же интерпретации некоторой итеративной программой (стандартной схемой), использующей* $2n + m + 2$ *переменных.*

Доказательство. Подробности даны в [5], а здесь мы ограничимся только псевдокодом эквивалентной (полуинтерпретированной) схемы итеративной программы:

1 : $var\ x, x_1, \ldots x_n :\ X;$
2 : $var\ y, y_1, \ldots y_n :\ Y;$
3 : $x := v;$
4 : $if\ p(x)\ then\ y := f(x)$
5 : $\quad else \left\{ do\ x := t_1(x)\ until\ p(x); \right.$
6 : $\quad\quad x_1 := x;\ x_2 := t_1(x_1);\ \ldots;\ x_n := t_1(x_{n-1});$
7 : $\quad\quad y_1 := f(x_1);\ y_2 := f(x_2);\ \ldots;\ y_n := f(x_n);$
8 : $\quad\quad do\ x := t_1^-(x);$
$\quad\quad\quad$ // Annotation: $x = t_1^-(x_1)$ & $bas(x) = \{x_1, \ldots\ x_n\}$ &
$\quad\quad\quad\quad$ // & $y_1 = G(x_1)$ & $\ldots$ & $y_n = G(x_n)$
9 : $\quad\quad\quad y :=\ g\Big(x,\ \big(h_1(x, y_1),\ \ldots,\ h_n(x, y_n)\big)\Big);$
10 : $\quad\quad\quad y_n := y_{n-1};\ \ldots;\ y_3 := y_2;\ y_2 := y_1;$
11 : $\quad\quad\quad y_1 := y;$
12 : $\quad\quad\quad x_1 := t_1^-(x_1);\ \ldots;\ x_n := t_1^-(x_n)$
13 : $\quad\quad until\ x = v \left. \right\}.$

□

## Заключение

В работе [5] можно найти обзор исследований по устранению рекурсии в интерпретированных программах, использованию (однократно выделяемых в динамической памяти) массивов для устранения рекурсии в рекурсивной схеме (1), примеры олимпиадных задач по математике и программированию, основанные на рекурсивной схеме (1).

Некоторые вопросы и направления для дальнейших исследований представлены ниже.

- Доказать с использованием компьютерных инструментов автоматического доказательства Теорему 1 (и другие утверждения из работе [5]).
- Исследовать, как обобщить рекурсивный шаблон (1) и условия Теоремы 1 таким образом, чтобы сохранить устранение рекурсии.
- Разработать и реализовать плагин для некоторой IDE (интегрированной среды разработки), который анализирует программный код для поиска рекурсивных шаблонов, допускающих устранение рекурсии.

## Список литературы

[1] *Котов, В. Е.* Теория схем программ / В. Е. Котов, В. К. Сабельфельд. — М. : Наука, 1991. — 274 с.

[2] *Легалов, А. И.* Преобразование хвостовых рекурсий в функционально-потоковых параллельных программах / А. И. Легалов, О. В. Непомнящий, И. В. Матковский, М. С. Кропачева // Моделирование и анализ информационных систем. — 2012. — Т. 19, № 4. — С. 48–58.

[3] *Шилов, Н. В.* Этюд об устранении рекурсии // Модел. и анализ информ. систем. — 2018. — Т. 25, № 5. — С. 549–560.

[4] *Knuth, D. E.* Textbook Examples of Recursion. — 1991. — 18 p. — URL: arXiv:cs/9301113. — Загл. с титул. экрана.

[5] *Shilov, N. V.* Teaching Efficient Recursive Programming and Recursion Elimination Using Olympiads and Contests Problems / N. V. Shilov, D. Danko // Frontiers in Software Engineering Education — First Int. Workshop, FISEE 2019. Lecture Notes in Computer Science. — V. 12271. — Springer, 2020. — P. 246–264.

**Библиографическая ссылка**

*Шилов, Н. В.* Устранение рекурсии в полуинтерпретированных схемах программ // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 85–90.
https://doi.org/10.26456/mfcsics-21-13

**Сведения об авторах**

Шилов Николай Вячеславович
Университет Иннополис. Доцент

*Россия, 420500, г. Иннополис, ул. Университетская, д. 1*
*E-mail: shiloviis@mail.ru*

УДК 510.62, 510.52
AMS MSC2020: 68N17

# Неподвижная точка для логических программ

**Авхимович Н. В.**

Тверской государственный университет

Аннотация. В данной работе рассматривается понятие частичной неподвижной точки для нормальных логических программ. Мы показываем, как при помощи неподвижной точки можно сократить запись логической программы, причем экспоненциально. Также исследуем задачу о вычислении неподвижной точки в общем случае и доказываем для нее PSPACE-полноту.

Ключевые слова: логическая программа, вычислительная сложность, неподвижная точка.

## Введение

Логические программы широко применяются для описания тех или иных процессов обработки информации [2]. Одним из методов применения логических языков является неподвижная точка.

В этой работе мы исследуем задачу построения частичной неподвижной точки для логических программ. Показываем, что в некоторых случаях, благодаря неподвижной точке, можно представлять информацию в более коротком виде.

Однако, задача о вычислении неподвижной точки для нормальных логических программ в общем случае, является более сложной: она полная в классе PSPACE.

## 1. Основные определения

Определение 1 (Нормальная логическая программа). *Нормальная логическая программа (см. [2]) — это конечное множество, состоящее из нормальных предложений, то*

*есть предложений вида*

$$y \leftarrow l_1 \wedge l_2 \wedge \ldots \wedge l_n,$$

*где $y$ — переменная без отрицания, а $l_i$ — переменная с отрицанием или без.*

В данной работе мы рассматриваем только нормальные логические программы. В дальнейшем, для краткости, будем называть их просто логическими программами, опуская слово нормальные.

Для логической программы $L$, все переменные, которые в ней встречаются, мы делим на две группы: входные и выходные. При наличии предложения

$$y \leftarrow l_1 \wedge l_2 \wedge \ldots \wedge l_n$$

в программе $L$ и истинности формулы $l_1 \wedge l_2 \wedge \ldots \wedge l_n$ на входных данных, считаем, что значение переменной $y$ должно быть равно единице. Если значение переменной $y$ не равняется единице ни в одном из предложений программы $L$, где $y$ стоит в левой части, то считаем его равным нулю.

Мы рассматриваем только стратифицированные логические программы, точнее, программы не содержащие циклов в определении значений переменных.

Определение 2 (Частичная неподвижная точка). *Набор $\bar{y}$ называется частичной неподвижной точкой программы $L$ на входе $\bar{x}$, если $L^i(\bar{x}) = L^{i+1}(\bar{x}) = \bar{y}$ для некоторого входного набора $\bar{x}$ и для некоторого натурального числа $i$.*

Рассмотрим следующую задачу (см. [3]). Пусть $\varphi(\bar{x}, \bar{y})$ — некоторая формула логики высказываний, находящаяся в дизъюнктивной нормальной форме. Пусть $\bar{x}$ и $\bar{y}$ — это наборы длины $n$. Построим по формуле $\varphi$ граф следующим образом: вершинами будут значения наборов $\bar{x}$ и $\bar{y}$, то есть наборы из нулей и единиц. Ребро между вершинами $\bar{x}$ и $\bar{y}$ существует тогда и только тогда, когда $\varphi(\bar{x}, \bar{y}) = 1$. Длина формулы $\varphi$ полиномиальна от $n$. Граф же имеет экспоненциальный размер относительно длины формулы $\varphi$.

Рассмотрим множество

$$\text{TRACK} = \{(\varphi, \bar{u}, \bar{v}) : \text{в графе, построенном по формуле } \varphi,$$

есть путь из $\bar{u}$ в $\bar{v}$ и из каждой вершины графа
исходит не более одного ребра}.

Известен следующий факт (см. [3]): множество TRACK является PSPACE-полным.

## 2. Основные результаты

Пусть у нас есть программа $K(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$:

$$y_1 \leftarrow x_1 \oplus \ldots \oplus x_n.$$

Построим программу $L(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$ следующим образом:

$$\begin{aligned} &y_1 \leftarrow x_1 \oplus x_n; \\ &y_i \leftarrow x_{i-1}, \quad i = \overline{3, n}. \end{aligned}$$

Теорема 1. *Существует натуральное число $m$ такое, что*

$$K(\bar{x}) = L^m(\bar{x}) = L^{m+1}(\bar{x})$$

*для любого набора $\bar{x}$.*

Таким образом, результат, выдаваемый программой $K$, можно получить при помощи построения неподвижной точки для программы $L$, то есть многократным ее применением. Оценим длины программ для обоих случаев. Поскольку мы рассматриваем только нормальные логические программы, необходимо представить $K$ и $L$ в нужном виде. Программа $K$ будет состоять из $2^{n-1}$ нормальных предложений. Программа $L$ будет состоять из $n$ нормальных предложений и будет применяться $n-1$ раз. Получаем:

$$\begin{aligned} |K| &= O(2^n); \\ |L^{n-1}| &= O(n^2). \end{aligned}$$

Неподвижная точка позволяет в некоторых случаях длинную программу записать коротко: например, это справедливо для программы экспоненциальной длины, приведенной в работе [1].

Если же разрешить использование нахождения частичной неподвижной точки произвольных программ, это может привести к большому времени работы.

Рассмотрим задачу о достижимости неподвижной точки

$$PFP = \{(L, \bar{x}, \bar{y}) : \bar{y} \text{ — частичная неподвижная точка логической программы } L \text{ для набора } \bar{x}\}$$

Теорема 2. *Задача $PFP$ является PSPACE-полной.*

## Заключение

Мы рассмотрели понятие частичной неподвижной точки для логических программ: показали как с помощью неподвижной точки можно сократить запись логической программы определенного вида, имеющей экспоненциальную длину, а также рассмотрели задачу о достижимости неподвижной точки в общем случае.

Было бы интересно найти более широкий класс логических программ, длины которых можно сократить за счет применения неподвижной точки.

## Список литературы

[1] *Avkhimovich, N.* Logic program invertibility in cryptography problems // Journal of Physics: Conference Series. — 2021. — Vol. 1902 — p. 012049.

[2] *Kifer, M.* Declarative Logic Programming: Theory, Systems, and Applications / M. Kifer, Y. A. Liu. — New York, N. Y. : Association for Computing Machinery and Morgan & Claypool, 2018. — 615 p.

[3] *Tantau, T.* A Note on the Complexity of the Reachability Problem for Tournaments // Electronic Colloquium on Computational Complexity. — 2001. — Vol. 8. — TR01-092.

## Библиографическая ссылка

*Авхимович, Н. В.* Неподвижная точка для логических программ // Всероссийская научная конференция «Математические основы ин-

форматики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 91–95.
https://doi.org/10.26456/mfcsics-21-14

**Сведения об авторах**

Авхимович Николь Вадимовна
Тверской государственный университет. Студент

*Россия, 170100, Тверь, ул. Желябова, 33*
*E-mail: nicoleavkhimovich@mail.ru*

УДК 004.81
AMS MSC2020: 68T99

# Метод максимального правдоподобия для обобщения нечетких множеств в таксономиях

**Айрапетян Ж. С., Фролов Д. С., Миркин Б. Г.**

НИУ «Высшая школа экономики»

Аннотация. В работе предлагается новый метод обобщения тематической текстовой коллекции, оснащенной таксономией предметной области. С помощью спектральных методов кластеризации из текстовой коллекции извлекаются нечеткие множества листьев таксономии, соответствующие понятиям, одновременно используемым в статьях коллекции. Эти нечеткие множества обобщаются путем их подъема в дереве таксономии с использованием критерия максимального правдоподобия. Оптимальный подъем подразумевает нахождение вершины или множества вершин в дереве таксономии, наиболее плотно покрывающих листовые понятия из обобщаемого множества. Наш метод включает два основных этапа: (1) извлечение кластеров из текстовой коллекции и (2) обобщение этих кластеров. В данной работе модернизируются оба этапа. Алгоритмы применены к структурному анализу и описанию текстовой коллекции из 17 тыс. аннотаций научных статей в области Наук о данных, опубликованных в журналах издательства Шпрингер. Таксономия Наук о данных, используемая в данной работе, является шестиуровневой иерархической таксономией, разработанной вручную международной Ассоциацией Вычислительной Техники и Вычислительных Систем (ACM-CSS [4])

Ключевые слова: иерархическая таксономия, методы обобщения, спектральная кластеризация, аннотированные суффиксные деревья.

## Введение

Вопросы автоматизации анализа текстовых коллекций приобретают все большее значение, как в силу практических потребностей, так

и в силу теоретической необходимости. В работе исследуется математический аналог уникальной когнитивной способности человека — обобщения. Понятие обобщения в данном контексте подразумевает извлечение концепций большего объема, но менее конкретного содержания из коллекции документов, то есть перехода от частного к общему согласно разработанному нами подходу [3]. Публикации в области анализа текстовых коллекций указывают на иерархическую структуру концепций. Такая иерархическая структура строится переходами от общего к частному, а значит, напрямую затрагивает понятие обобщения. Таксономия предметной области представляет собой иерархическую структуру корневого дерева. Первым делом из текстовой коллекции извлекаются нечеткие множества листьев таксономии, отражающие структуру текстовой коллекции, а затем эти множества обобщаются с использованием структуры таксономии. Такая процедура позволяет выявить узлы таксономии, наиболее точно описывающие текстовую коллекцию.

## 1. Извлечение кластеров

Для извлечения нечетких кластеров на множестве листьев таксономии необходимо вычислить релевантность каждого текста к каждой листовой теме таксономии. Для вычисления матрицы релевантностей $R$ используются Аннотированные Суффиксные деревья (AST [2]), построенные для каждого текста из коллекции. Матрица ко-релевантности листовых тем $A$ вычисляется как $A = R^T R$, то есть для двух листовых тем $i$ и $j$ с векторами релевантности текстам $r_i$ и $r_j$ их схожесть определяется как скалярное произведение. Мы также используем веса текстов, учитывающие их уникальность.

Теперь задачу кластеризации можно формулировать в терминах оптимального разбиения взвешенного графа, представляемого матрицей $A$, на кластеры. Такую задачу можно решить с помощью комбинации двух алгоритмов: LaplacianEigenMaps [1] и метода нечетких $C$-средних. Матрица смежности $A$ используется для вычисления матрицы нормализованного Лапласиана $L_n$ , спектр которого (особенно собственные векторы, соответствующие наименьшим собственным значениям) имеет интерпретацию в терминах минимального числа разрезов, необходимого для разделения графа на компоненты сопо-

ставимого размера. Затем полученные вложения кластеризуются с помощью метода нечеткой кластеризации $C$-средних.

## 2. Методы обобщения

После извлечения нечетких кластеров из текстовой коллекции, мы хотим обобщить эти кластеры, используя дерево таксономии. Введем некоторые обозначения и определения для удобства дальнейших рассуждений. Пусть $I$ — множество листьев нашей таксономии $T$, где $T$ — множество всех узлов таксономии, тогда для любого $h \in T \setminus I$: $T(h)$ будет поддеревом дерева $T$, а $I(h)$ — терминальными узлами этого поддерева. Для любого $h \in T \setminus I$ будем обозначать множество непосредственных отпрысков узла $h$ как $\chi(h)$. Дадим определение нечеткому множеству. Нечеткое множество на листьях $I$ нашего дерева — это отображение $u : I \to \mathbb{R}^{[0,1]}$, где $u(i)$ для $i \in I$ — это значение принадлежности листового элемента данному множеству. $S_u = \{i \in I : u(i) > 0\}$ — основа нечеткого множества $u$. Все вершины $t \in T$, такие что $I(t) \cap S_u = \varnothing$ будем называть $u$-нерелевантными вершинами. Из такого определения вытекает, что для любой $u$-нерелевантной вершины ее потомки тоже являются $u$-нерелевантными. Максимально $u$-нерелевантным узлом называется узел, который является $u$-нерелевантным, а его родитель уже не является $u$-нерелевантным. Удалим из дерева таксономии все не максимальные $u$-нерелевантные узлы. Теперь все $u$-нерелевантные узлы — это листья, не входящие в основу нечеткого множества. В работе [3] была рассмотрена следующая задача обобщения: дано нечеткое множество  на терминальных узлах таксономии $T$, требуется найти вершину $h \in T$, которая покрывала бы данное множество настолько плотно, насколько это возможно. Набор узлов $H$ будем называть $u$-покрытием, если:

(а) $H$ покрывает $S_u$, то есть $S_u \subseteq \bigcup\limits_{h \in H} I(h)$;

(б) узлы в $H$ не связаны, то есть $I(h) \cap I(h') = \varnothing$ для любых $h, h' \in H$ таких, что $h \neq h'$.

Узлы, принадлежащие $H \setminus I$, будут являться головными темами покрытия, а узлы, принадлежащие $H \cap I$, индуцированными ею

выбросами. Множеством пробелов покрытия будет являться объединение по всем индуцированным покрытием $u$-нерелевантым узлам, то есть $\bigcup_{h\in H\setminus I} I(h)\setminus S_u$. Обозначим $G(h)=I(h)\setminus S_u$. Значение штрафной функции, ассоциированное с определенным $u$-покрытием должно учитывать принадлежности $u(i)$ листовых элементов своих головных тем, поэтому чтобы корректно определить штрафную функцию для $u$-покрытия нужно расширить принадлежность листовых элементов на все узлы дерева. Будем считать, что принадлежности листовых элементов уже нормированы и $\sum_{i\in I} u(i)^2 = 1$. Выберем следующую функцию агрегации для внутренней вершины $t$: $u(t)=\sqrt{\sum_{i\in I(t)} u(i)^2}$. Итак, ассоциированная с $u$-покрытием $H$ штрафная функция, которая учитывает важность головных тем, пробелов и выбросов с соответствующими весами 1, $\lambda$ и $\gamma$, будет выглядеть следующим образом:

$$p(H)=\sum_{h\in H\setminus I} u(h)+\lambda\sum_{h\in H\setminus I}\sum_{g\in G(h)} v(g)+\gamma\sum_{h\in H\cap I} u(h), \qquad (1)$$

где $\lambda$ и $\gamma$ — гиперпараметры. В статье [3] построен рекурсивный алгоритм, находящий глобальный минимум данной функции. В данной работе мы модифицируем этот метод, исходя из вероятностей приобретения и потерь головных тем в узлах. Данный метод позволяет избавиться от гиперпараметров $\lambda$ и $\gamma$.

Для оценки априорных вероятностей потерь и приобретений для всех узлов, мы применили многократный запуск алгоритма с критерием максимальной экономии (1) примерно 300 раз на случайных наборах по 5000 статей из текстовой коллекции. Для нахождения глобально оптимального решения по критерию максимального правдоподобия используется рекурсивный алгоритм, схожий с рекурсивным алгоритмом для критерия максимальной экономии [3]. Применение нового метода позволяет уточнить и дополнить ранее полученные результаты о тенденциях исследований в области науки о данных.

В работе принимали участие Сузана Насименто (Новый университет Лиссабона, Португалия) и Тревор Феннер (Университет Биркбек, Лондон Великобритания).

**Список литературы**

[1] *Belkin M.* Laplacian eigenmaps for dimensionality reduction and data representation / M. Belkin, P. Niyogi // Neural Computation. — 2003. — Vol. 15, №6. — P. 1373–1396.

[2] *Chernyak, E.* Refining a taxonomy by using annotated suffix trees and wikipedia resources / E. Chernyak, B. Mirkin // Annals of Data Science. — 2015. — Vol. 2, №1. — P. 61–82.

[3] *Frolov, D.* Parsimonious Generalization of Fuzzy Thematic Sets in Taxonomies Applied to the Analysis of Tendencies of Research in Data Science / D. Frolov, S. Nascimento, T. Fenner, B. Mirkin // Information Sciences. — 2020. — Vol. 512. — P. 595–615.

[4] The 2012 ACM Computing Classification System. — URL: https://www.acm.org/publications/class-2012. — Загл. с титул. экрана.

**Библиографическая ссылка**

*Айрапетян, Ж. С.* Метод максимального правдоподобия для обобщения нечетких множеств в таксономиях / Ж. С. Айрапетян, Д. С. Фролов, Б. Г. Миркин // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 96–101. https://doi.org/10.26456/mfcsics-21-15

**Сведения об авторах**

1. Айрапетян Жирайр Сережаевич
НИУ «Высшая школа экономики». Исследователь в лаборатории ИССА

*Россия, Москва, Покровский бульвар, д.11*
*E-mail: zhayrapetyan@ithse.ru*

2. Фролов Дмитрий Сергеевич
НИУ «Высшая школа экономики». Исследователь МЦАВР

*Россия, Москва, Покровский бульвар, д.11*
*E-mail: dmitsf@gmail.com*

3. Борис Григорьевич Миркин
НИУ «Высшая школа экономики». профессор

*Россия, Москва, Покровский бульвар, д.11*
*E-mail: bmirkin@hse.ru*

УДК 69.059
AMS MSC2020: 60K25

# Системы с перерывами обслуживания и их применения[1]

**Афанасьев Г. А.**

Московский государственный строительный университет

Аннотация. Рассматривается одноканальная система обслуживания с перерывами в работе прибора, когда он освобождается от обслуживания требований. Перерыв прерывается, если число требований в системе достигает некоторого фиксированного уровня. Входящий поток требований — пуассоновский. В общих предположениях относительно распределений случайных величин, определяющих функционирование системы, находятся стационарное распределение числа требований в системе, его среднее значение, а также изучается асимптотика числа завершенных перерывов за большое время. Обсуждается применение полученных результатов в организации деятельности управляющей компании техническим комплексом жилых зданий.

Ключевые слова: системы обслуживания, перерывы в обслуживании, стационарное распределение.

Рассматривается одноканальная система с перерывами в обслуживании $S_m$ с пуассоновским входящим потоком $X$ интенсивности $\lambda$ и независимыми временами обслуживания с функцией распределения $B(x)$ и средним $b < \infty$. Когда система освобождается от обслуживания всех имеющихся в ней требований, прибор переключается на другой режим работы на случайное время, которое мы называем перерывом. Если система свободна в конце перерыва, начинается новый перерыв. В противном случае прибор приступает к обслуживанию требований в стандартном режиме. В течение перерыва в систему продолжают поступать требования и в момент, когда в ней будет $m$ требований, перерыв обрывается и начинается обслуживание в стандартном режиме.

[1] Работа выполнена при частичной поддержке гранта РФФИ (проект №20-01-00-487)

Системы с перерывами в обслуживании привлекали многих исследователей, а сама идея перерывов возникла в работе [4]. Широкий обзор моделей с перерывами дан в статьях [5–7]. За последние годы предложено много новых моделей систем с перерывами, что в первую очередь связано с широким кругом приложений в различных сферах [1–3, 7].

Сначала рассмотрим систему без прерывания перерывов, то есть $S_\infty$. Пусть $\{T_n\}_{n=0}^{\infty}$ — последовательные моменты, в которые начинаются перерывы, а $Y_n(t)$ — число требований в системе в момент $T_n + t$ и $\eta_n$ — длительность $n$-го перерыва. Предполагается, что последовательность $\{Y_n(t), \eta_n, t \leqslant \eta_n\}_{n=1}^{\infty}$ состоит из независимых одинаково распределенных случайных элементов, не зависящих от времени обслуживания. Считаются заданными функции

$$V(z,t) = Ez^{Y(t)} \parallel (\eta > t) = \sum_{j=0}^{\infty} z^j P(Y(t) = j,\ \eta > t),$$
$$G(z,s) = Ez^{Y(t)} e^{-s\eta},\ f(s) = G(1,s) = Ee^{-s\eta}, \tag{1}$$
$$C(z) = Ez^{Y(\eta)} = G(z,0),\ |z| \leqslant 1,\ Res \geqslant 0.$$

Мы изучаем два процесса: $q(t)$ — число требований в системе $S_\infty$ и $n(t)$ — число завершенных перерывов к моменту $t$. Заметим, что процесс $q(t)$ стабилен тогда и только тогда, когда $\rho = \lambda b < 1$ (см. теорема 1 в [1]). Предельное распределение этого процесса дается формулой (теорема 2 в [1])

$$\lim_{t\to\infty} Ez^{q(t)} = \pi(z) = \frac{1-\rho}{\lambda\bar{\eta}(1-\rho) + \rho Y_1} \times$$
$$\times \left( \lambda \int_0^{\infty} V(z,t)dt + \frac{z(1-C(z))}{1-z} \cdot \frac{1-\beta(\lambda-\lambda z)}{\beta(\lambda-\lambda z) - z} \right), \tag{2}$$

где $\bar{\eta} = E\eta,\ Y_1 = EY(\eta),\ \beta(s) = \int_0^{\infty} e^{-sx} dB(x)$.

Дифференцируя (2) по $z$ и полагая $z = 1,\ b_2 = \int_0^{\infty} x^2 dB(x) < \infty$,

$E\eta^2 < \infty$, $EY^2(\eta) = Y_2 < \infty$, мы имеем (теорема 3 в [1])

$$Eq = \pi'(1) = (\lambda\bar{\eta}(1-\rho) + \rho Y_1)^{-1} \left[ (1-\rho)\lambda \times \right.$$
$$\left. \times \int_0^\infty EY(t) \parallel (\eta \geqslant t) dt + \rho Y_1 \left( 1 + \frac{Y_2 - Y_1}{2Y_1} + \frac{\lambda^2 b_2}{2\rho(1-\rho)} \right) \right]. \quad (3)$$

Для процесса $n(t)$ в системе $S_\infty$ мы имеем следующие асимптотические результаты.

1) Существует
$$\lim_{t\to\infty} \frac{EN(t)}{t} = \frac{1-\rho}{\bar{\eta}(1-\rho) + bY_1}. \quad (4)$$

2) Если $E\eta^2 < \infty$, $EY^2(\eta) < \infty$, $b_2 = \int_0^\infty x^2 dB(x) < \infty$, то

$$P\left( \frac{n(t) - \dfrac{t}{E\tau}}{\dfrac{\sqrt{t}\sigma_\tau}{(E\tau)^{3/2}}} < \infty \right) \xrightarrow[t\to\infty]{} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{\frac{-y^2}{2}} dy = \Phi(x). \quad (5)$$

Здесь

$$E\tau = \bar{\eta} + \frac{b}{1-\rho} Y_1,$$
$$\sigma_\tau^2 = \frac{b^2}{(1-\rho)^2} DY(\eta) + D\eta + \frac{2b}{1-\rho} \cdot E\left(Y(\eta) - Y_1\right)(\eta - \bar{\eta}) +$$
$$+ \frac{Y_1}{(1-\rho)^3} \left( b_2 - (1-\rho)b^2 \right). \quad (6)$$

Теперь рассмотрим систему $S_m$ для $m < \infty$, предположив, что перерыв в обслуживании прерывается в момент $t_m$, когда число требований в системе становится равным $m$. Если длительность перерыва $\eta < t_m$, то он не прерывается. Дополнительно предположим,

что последовательность $\{Y_n(t), t \geqslant 0\}_{n=1}^{\infty}$ и $\{\eta_n\}_{n=1}^{\infty}$ независимы и обозначим $t_m = \min\{t \geqslant 0 : Y(t) = m\}$, $g_m(x) = P(t_m < x)$, $\eta^{(m)} = \min\{\eta, t_m\}$, $\varphi_m(z,t) = Ez^{Y(t)} \parallel (t_m > t)$.

Все характеристики системы $S_m$ пометим буквой $m$. Если $F(t) = P(\eta \leqslant \tau)$, то

$$\bar{\eta}^{(m)} = E\eta^{(m)} = \int_0^{\infty} x\,(1 - g_m(x))\,dF(x) + \int_0^{\infty} x\,(1 - F(x))\,dg_m(x), \quad (7)$$

$$V_m(z,t) = (1 - F(t))\,\varphi_m(z,t), \quad (8)$$

$$C_m(z) = \int_0^{\infty} (\varphi_m(z,x) + z^m g_m(x))\,dF(x). \quad (9)$$

Поскольку $Y_1^{(m)} = C_m'(1)$ $Y_2^{(m)} = C_m''(1) + C_m'(1)$, можно получить формулы для $Y_1^{(m)}$ и $Y_2^{(m)}$, используя (9). Теперь, взяв вместо $\bar{\eta}, C(z), V(z,t)$ соответственно $\bar{\eta}^{(m)}, C_m(z)$, $V_m(z,t)$, полученные по формулам (7), а вместо $Y_1$ и $Y_2$ — величины $Y_1^{(m)}$ и $Y_2^{(m)}$ в формулах (2) и (3), мы найдем предельное распределение $\pi_m(z) = \lim\limits_{t\to\infty} Ez^{q_m(t)}$ процесса $q_m(t)$ в системе $S_m$. Аналогично находится асимптотика процесса $n_m(t)$.

В качестве примера рассмотрим систему $S_m$ в предположении, что $Y_m(t)$ — пуассоновский процесс интенсивности $\mu$, а $\eta$ имеет экспоненциальное распределение с параметром $\nu$. Осуществив описанные действия, мы получаем следующие результаты. Если $\rho = \lambda b < 1$ и $\alpha = \frac{\mu}{\mu+\nu}$, то для системы $S_m$

$$\pi_m(z) = \frac{(1-\rho)\nu\,(1-\alpha^m z^m)}{(1-\alpha^m)\,(\lambda(1-\rho)+\rho\mu)\,(\nu+\mu(1-z))} \times \\ \times \left(\lambda + \frac{\mu z(1-\beta(\lambda-\lambda z))}{\beta(\lambda-\lambda z) - z}\right)$$

и

$$Eq_m = (\lambda(1-\rho)+\rho\mu)^{-1} \times \\ \times \left(\frac{\lambda\mu}{\nu}(1-\rho) + \frac{\rho\mu}{1-\alpha} + \frac{\lambda^2 b_2 \mu}{2(1-\rho)} - \frac{m\alpha^m\,(\lambda(1-\rho)+\rho\mu)}{1-\alpha^m}\right).$$

Аналогично с помощью (5) и (6) получаются формулы для нормирующих коэффициентов $E\tau^{(m)}$ и $\sigma^2_{\tau^{(m)}}$ процесса $n_m(t)$.

Системы с перерывами в обслуживании могут быть использованы для изучения многих прикладных моделей. Здесь мы рассмотрим применение полученных результатов в организации деятельности управляющей компании (УК) техническим комплексом жилых зданий. Подобный подход уже использовался в работах [2, 3]. Будем учитывать две важнейшие задачи УК — предупредительные инспекции и ремонты и устранение внезапных поломок технического оборудования. Считается, что предупредительная инспекция и ремонт может начаться, когда нет экстренных вызовов, связанных с внезапными поломками. Тогда $q(t)$ — это количество экстренных вызовов, имеющихся в момент $t$, а $n(t)$ — число предупредительных ремонтов, осуществленных за время $(0, t)$. Цели УК, с одной стороны обеспечить требуемое количество этих ремонтов, а с другой не допустить большой очереди из экстренных запросов. Мы предполагаем, что плановые профилактические ремонты и осмотры обрываются, когда число экстренных вызовов достигает уровня $m$. В качестве математической модели используется система $S_m$. При этом перерыв в обслуживании означает, что бригада специалистов занимается профилактическим осмотром и ремонтом оборудования, а работа в стандартном режиме — это обслуживание экстренных вызовов. Имея достаточное число экспериментальных данных, можно получить оценки для основных параметров модели, а затем по формулам (3), (5), (6) найти среднее число ожидающих экстренных запросов и распределение числа завершенных плановых ремонтов и осмотров.

**Список литературы**

[1] *Афанасьев, Г. А.* Система M|G|1 с перерывами в работе и их задержками // Теория вероятностей и ее применения. — 2021. — Т. 66, вып. 1. — С. 3–19.

[2] *Афанасьев, Г. А.* Использование теории массового обслуживания для организации эксплуатации инженерных систем жилых зданий // Вестник ТвГУ. Серия: Прикладная математика. — 2019. — №4. — С. 52–64.

[3] *Afanasyev, G. A.* Scheduling prophylactic maintenance of engineering systems of residential buildings / G. A. Afanasyev, K. A. Shreiberg // Journal of Physics: Conference Series. — 2019. — Vol. 1425. — ID 012045.

[4] *Levy, Y.* Utilization of Idle Time in an M/G/1 queueing system / Y. Levy, U. Yechiali // Management Science. — 1975. — Vol. 22, №2. — P. 202–211.

[5] *Niu, Z.* A vacation queue with setup and close-down times and hatch Marcovian arrival processes / Z. Niu, T. Shu, Y. Takahashi // Performance Evaluation. — 2003. — Vol. 54, №3. — P. 225–248.

[6] *Srernivasang, C.* MAP/PH/1 queue with working vacations, vacation interruptions and N policy / C. Srernivasang, S. R. Chachavarthy, A. Krishnamoorthy // Applied Mathematical Modelling. — 2013. — Vol. 37, №6. — P. 3879–3893.

[7] *Zhang, M.* Performance analysis of M/G/1 queue with working vacations and vacation interruption / M. Zhang, Z. Hou // Journal of Computational and Applied Mathematics. — 2010. — Vol. 234, №10. — P. 2977–2985.

### Библиографическая ссылка

*Афанасьев, Г. А.* Системы с перерывами обслуживания и их применения // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 102–107.
https://doi.org/10.26456/mfcsics-21-16

### Сведения об авторах

Афанасьев Григорий Александрович
Московский государственный строительный университет. Доцент

*Россия, 129337, г. Москва, Ярославское шоссе, д. 26*
*E-mail: gregoria@mail.ru*

УДК 519.218
AMS MSC2020: 60K25

# Асимптотический анализ систем обслуживания с повторными вызовами при регенерирующем входящем потоке[1]

**Афанасьева Л. Г., Баштова Е. Е.**

Московский государственный университет им. М. В. Ломоносова

Аннотация. Рассматривается многоканальная система с повторными вызовами и постоянной интенсивностью запросов с орбиты. Времена обслуживания требований имеют произвольное распределение, а входящий поток предполагается регенерирующим. На основе метода синхронизации и теорем о сильной гауссовской аппроксимации регенерирующих потоков мы устанавливаем аналог сильного принципа инвариантности Штрассена для количества требований в перегруженной системе.

Ключевые слова: системы обслуживания, повторные вызовы, условия стабильности, сильный принцип инвариантности.

## Введение

Системы с повторными вызовами изначально представляли собой альтернативу классическим моделям телефонных систем, а именно систем с отказами, которые не учитывают возможности повторения вызовов от требований, получивших отказ. В системах с повторными вызовами можно рассматривать различные правила формирования потока вызовов с орбиты. Если предположить, что каждое блокированное требование независимо от других повторяет запросы через экспоненциально распределенные промежутки времени, то мы получим классическую политику повторов. Другой класс включает в себя системы с постоянной интенсивностью повторов. Эти системы

[1]Работа выполнена при финансовой поддержке РФФИ, проект 20-01-00487.

возникли в работе [7] как модели телефонной связи. Далее произошел быстрый рост литературы (см. [4], [6] и литературу там), что в первую очередь связано с успешным применением систем этого класса при анализе коммуникационных и компьютерных сетей, где попытки повторов осуществляются процессором независимо от числа сообщений, хранящихся в узлах сети.

В работе [1] получено условие эргодичности для систем с повторными вызовами. В настоящей работе мы изучаем асимптотическое поведение количества требований в перегруженных системах и доказываем аналог сильного принципа инвариантности Штрассена для процесса, описывающего количество требований в системе.

## 1. Описание модели и основной результат

Имеется $m$ одинаковых приборов. Времена обслуживания на каждом приборе — независимые одинаково распределенные случайные величины (н. о. р. с. в.) с функцией распределения $B(x)$. Если в момент поступления требования есть хотя бы один свободный прибор, то требование сразу начинает обслуживаться. Если же все приборы заняты, то оно отправляется на так называемую орбиту, откуда повторяет попытки попасть на обслуживание. Запросы с орбиты поступают через н. о. р. интервалы. Если в момент запроса есть свободный прибор, а на орбите есть требования, то одно из них отправляется на обслуживание. Такая модель называется моделью с постоянной интенсивностью запросов с орбиты.

Мы предполагаем, что входящий поток $X(\cdot)$ является регенерирующим.

Определение 1 (Регенерирующий поток). *Процесс $X(t)$ с неубывающими, непрерывными справа и имеющими предел слева траекториями называется регенерирующим потоком, если существует возрастающая последовательность $\{\theta_i, i \geqslant 0\}$, $\theta_0 = 0$ такая, что последовательность*

$$\{\kappa_j\}_{j=1}^{\infty} = \{X(\theta_{j-1}+t) - X(\theta_{j-1}), \theta_j - \theta_{j-1}, t \in [0, \theta_j - \theta_{j-1})\}_{j=1}^{\infty}$$

*состоит из н. о. р. случайных элементов.*

При этом случайная величина $\theta_j$ называется $j$-й точкой регенерации $X(t)$, а $\tau_j = \theta_j - \theta_{j-1}$ — $j$-м периодом регенерации. Основные

свойства регенерирующих потоков приведены в [3]. Одним из них является тот факт, что если периоды регенерации $\tau_j$ и приращения $\xi_j$ процесса $X(t)$ на этих периодах имеют конечное математическое ожидание, то существует интенсивность $\lambda = \lim\limits_{t\to\infty} X(t)/t$.

Мы изучаем процесс $Q(t)$ — число требований в системе (на орбите и на приборах вместе).

Введем дополнительное условие на входящий поток и времена обслуживания.

Условие 1. $\mathsf{P}(\xi_1 = 0, \tau_1 > 0) + \mathsf{P}(\xi_1 = 1, \tau_1 - t_1 > \eta_1) > 0$ *(здесь* $t_1$ *— момент поступления первого требования в систему).*

Пусть $\{\zeta_n\}_{n=1}^{\infty}$ — последовательность н. о. р. с. в., представляющих собой интервалы между запросами с орбиты, $N(t)$ — процесс восстановления, построенный по последовательности $\{\zeta_n\}_{n=1}^{\infty}$, его интенсивность $\nu = \lim\limits_{t\to\infty} N(t)/t$.

Условие 2. *Случайные величины* $\zeta_n$, $n \in \mathrm{N}$ *имеют экспоненциальную фазу, то есть* $\zeta_n = \zeta_n^{(1)} + \zeta_n^{(exp)}$, $\zeta_n^{(1)}$ *и* $\zeta_n^{(exp)}$ *независимы, и* $\zeta_n^{(exp)}$ *имеют экспоненциальное распределение.*

Для формулировки результатов необходимо ввести вспомогательную систему $M_0$ с $m$ приборами, входящим потоком $U(t) = X(t) + N(t)$ и отказами. Обозначим $n(t)$ — число занятых приборов в $M_0$, а $t_k^0$ — $k$-й момент поступления требования в $M_0$. В статье [1] показано, что при выполнении условий 1 и 2 существуют пределы

$$\lim_{k\to\infty} \mathsf{P}(n(t_k^0) = j) = P_j, \quad j = \overline{1, m},$$

и коэффициент загрузки системы равен

$$\rho = \frac{\lambda}{(\lambda + \nu)(1 - P_m)},$$

а также доказана следующая эргодическая теорема.

Теорема 1. *Пусть выполнены условия 1 и 2.*
*Если* $\rho > 1$, *то*

$$Q(t) \xrightarrow{P} \infty \quad \text{при } t \to \infty.$$

*Если* $\rho < 1$, *то* $Q(t)$ *стабилен, то есть при любом начальном состоянии существует*

$$\lim_{t\to\infty} \mathsf{P}(Q(t) \leqslant x) = \Phi(x),$$

*где $\Phi(x)$ — функция распределения, не зависящая от начального состояния.*
*Если $\rho = 1$ и времена обслуживания также имеют экспоненциальную компоненту, то*

$$Q(t) \xrightarrow{P} \infty \quad \textit{при} \quad t \to \infty.$$

Основной результат нашего доклада — сильная гауссовская аппроксимация длины очереди в ситуации, когда система перегружена.
Теорема 2. *Пусть $\rho > 1$, выполнены условия 1 и 2 и $\mathsf{E}\tau_i^r < \infty$, $\mathsf{E}\xi_i^r < \infty$, $\mathsf{E}\zeta_i^r < \infty$, для $r > 2$. Тогда на некотором вероятностном пространстве $(\Omega, \mathcal{F}, \mathsf{P})$ можно одновременно определить процесс $Q$, имеющий распределение числа требований в системе, и стандартный Винеровский процесс $W$ так, что для некоторой константы $\sigma^2 > 0$*

$$\sup_{0 \leqslant u \leqslant t} \|Q(u) - (\rho - 1)u - \sigma W(u)\| = o(t^{1/r}), \ \textit{п. н.},$$

*при $t \to \infty$.*

Доказательство основано на методе синхронизации [2], подходящем мажорировании, теоремах о гауссовской аппроксимации регенерирующих потоков [5] и дополнительной технике.

## Заключение

Представленная теорема об аппроксимации с вероятностью единица процесса длины очереди имеет разнообразные следствия, такие как, например, сходимость в пространстве Скорохода и закон повторного логарифма. Кроме того, эта теорема позволит в дальнейшем, на основе наблюдения количества требований в системе, получить состоятельную оценку параметра $\sigma^2$, строить доверительные интервалы и проверять гипотезы о величине коэффициента загрузки, что является важным для приложений.

## Список литературы

[1] *Афанасьева, Л. Г.* Условия стабильности системы с повторными вызовами при регенерирующем входящем потоке // Фундаментальная и прикладная математика. — 2018. — Т. 22, № 3. — С. 5–18.

[2] *Afanasyeva, L. G.* Asymptotic Analysis of Queueing Models Based on Synchronization Method // Methodology and Computing in Applied Probability. — 2020. — Vol. 22. — P. 1417–1438.

[3] *Afanasyeva, L. G.* Coupling method for asymptotic analysis of queues with regenerative input and unreliable server / L. G. Afanasyeva, E. E. Bashtova // Queueing Systems. — 2014. — Vol. 76. — P. 125–147

[4] *Artalejo, S. R.* Analysis of multiserver queues with constant retrial rate / S. R. Artalejo, A. Gómez-Corral, M. F. Neuts // European Journal of Operational Research. — 2001. — Vol. 135. — P. 569–581.

[5] *Bashtova, E.* Strong Gaussian approximation for cumulative processes with heavy tails / E. Bashtova, A. Shashkin. — URL: arXiv:2007.15481. — Загл. с титул. экрана.

[6] *Falin, G. I.* Retrial Queues / G. I. Falin, J. G. C. Templeton. — London : Chapman & Hall, 1997. — 320 p.

[7] *Fayolle, G.* A simple telephone exchange with delayed feedback // Proc. of the international seminar on Teletraffic Analysis in Computer Performance Evaluation / Eds. O. S. Boxma, S. W. Cohen, H. C. Tijms. — Amsterdam : Elsevier, 1986. — P. 245–253.

### Библиографическая ссылка

*Афанасьева, Л. Г.* Асимптотический анализ систем обслуживания с повторными вызовами при регенерирующем входящем потоке / Л. Г. Афанасьева, Е. Е. Баштова // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 108–113.
https://doi.org/10.26456/mfcsics-21-17

### Сведения об авторах

1. Лариса Григорьевна Афанасьева
Московский государственный университет им. М. В. Ломоносова. Профессор

*Россия, 119991, г. Москва, Ленинские Горы, 1*
*E-mail: l.g.afanaseva@yandex.ru*

2. Елена Евгеньевна Баштова
Московский государственный университет им. М. В. Ломоносова. Доцент

*Россия, 119991, г. Москва, Ленинские Горы, 1*
*E-mail: bashtovaelenae@gmail.com*

УДК 519.7
AMS MSC2020: 68Q85

# Вопрос о графах достижимости сетей Петри

**Белов Ю. А.**

Ярославский государственный университет им. П. Г. Демидова

Аннотация. Ставится вопрос о возможности моделирования некоторого графа с помощью графа достижимости какой-либо сети Петри.

Ключевые слова: сети Петри, граф достижимости сети Петри, изоморфизм графов.

## Введение

Допустимые последовательности имен переходов данной сети образуют формальный язык в алфавите $T$ (список имен переходов) — так называемый свободный язык сети Петри. Если переходы помечены символами некоторого другого алфавита $A$, данные последовательности преобразуются в слова в алфавите $A$ и получается префиксный язык сети. При этом можно ограничиваться только такими допустимыми цепочками, которые заканчиваются в данном фиксированном «терминальном» состоянии. Таким образом получаются свободные или префиксные терминальные языки.

Обобщая ситуацию далее, можно считать некоторые переходы невидимыми и получать соответствующие языки с невидимыми переходами. Комбинируя перечисленные свойства в различных сочетаниях получаем семейство из шести типов языков, сопоставляемых данной сети Петри. Это дает возможность изучения «языковой выразительной мощности» сети, и сравнения различных типов языков. Указанное направление исследований является классическим в теории сетей Петри, многие материалы изложены, например, в [1]. Известно, в частности, что все регулярные языки моделируются, некоторые контекстно-свободные языки не моделируются сетями

Петри, в то же время существуют языки, моделируемые сетью Петри, но не являющиеся контекстно-свободными и т. п.

Если рассматривать проблему достижимости, видимо, требуется выяснять строение множества всех достижимых состояний данной сети. Это множество образует ориентированный граф с выделенным (начальным) состоянием. Если какой-то ориентированный граф изоморфен графу допустимости некоторой сети, граф можно назвать моделируемым сетью Петри. Аналогично языковому подходу можно привести примеры графов, не моделируемых сетями Петри, а также указать моделируемые графы.

Все конечные графы моделируются, и даже автоматными сетями Петри — предложение 1.

Конечно, моделируемый граф может, вообще говоря, моделироваться различными сетями, например, сетями различных размерностей. В частности, возникает вопрос минимизации такой сети — до некоторой степени аналог вопроса минимизации конечного автомата для регулярных языков и построение соотвествующего алгоритма минимизации. При этом требуется уточнять типы сетей, форму их задания и т. п.

Для бесконечных графов можно указать пример не моделируемого графа — однородное свободное дерево степени $d$, если $d > 2$. Для данного дерева количество вершин уровня $k$ (уровень — расстояние от корня до вершины) имеет порядок $(d-1)^k$, что используется при доказательстве отсутствия моделируемости. Точнее, используется следующее замечание: если в бесконечном ориентированном графе количество вершин уровня $k$ растет по экспоненте, то такой граф не моделируем, так как можно доказать, что в моделируемом графе указанное количество вершин имеет скорость роста не более степенной. Последнее утверждение (в языковой формулировке) можно снова найти в [1]. В предложении 2 дается некоторое усиление этого замечания: имеется пример счетного бесконечного дерева, в котором количество вершин уровня $k$ всегда равно 1 или 2 (в зависимости от $k$), степень каждой вершины равна 2 или 3 и это дерево не моделируемо.

Повторим точное определение. Ориентированный граф $G$ с выделенной вершиной назовем моделируемым, если существует такая сеть Петри $P$, граф достижимости которой [1], изоморфен данному графу $G$.

Отметим, что при установлении изоморфизма рассматривается только наличие или отсутствие дуги, соединяющей две вершины, а имена или метки дуг не учитываются.

## 1. Результаты

Предложение 1. *Каждый конечный ориентированный граф $G$ с отмеченной вершиной, без петель и параллельных дуг, является моделируемым.*

При этом сеть Петри даже можно считать консервативной, в которой циркулирует только одна фишка.

Предложение 2. *Существует ориентированное бесконечное счетное выходящее корневое дерево, в котором вершин $k$-го уровня имеется имеется ровно одна или две для любого $k$, и такое, что данное дерево не моделируемо.*

## Заключение

Вопрос о достаточных условиях моделируемости бесконечных графов, видимо, является открытым.

## Список литературы

[1] *Котов, В. Е.* Сети Петри. — М. : Наука, 1984. — 156 с.

## Библиографическая ссылка

*Белов, Ю. А.* Вопрос о графах достижимости сетей Петри // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 114–117.
https://doi.org/10.26456/mfcsics-21-18

**Сведения об авторах**

Юрий Анатольевич Белов
Ярославский государственный университет им. П. Г. Демидова. Доцент

*Россия, 150003, г. Ярославль, ул. Советская, д. 14*
*E-mail: belov45@yandex.ru*

УДК 004.89
AMS MSC2020: 68Q45

# Комплекс алгоритмов Data Mining в исследовании процесса протекания химических реакций

**Биллиг В. А., Звягинцев Н. В.**

Тверской государственный технический университет

Аннотация. В настоящее время накоплено значительное количество экспериментальных данных, фиксирующих процесс протекания химических реакций. Анализ этих данных комплексом алгоритмов Data Mining дает важную практическую информацию для поиска эффективных условий проведения реакций, при которых получается максимальное количество целевого продукта при минимальных затратах. В данной работе на примере работы с базой, содержащей данные о протекании реакции карбонилирования различных олефинов, показано, как разработанный нами программный комплекс, позволяет извлечь полезные знания, способствующие повышению эффективности протекания химических реакций.

Ключевые слова: Data Mining, Apriori, деревья решений, химические реакции.

## Введение

Химические реакции (ХР) являются сложными процессами, на протекание которых оказывает влияние масса условий: давление, температура, состав и природа взаимодействующих веществ и катализаторов. Вместе с тем, важной задачей является поиск наиболее эффективных условий, когда при минимальных затратах получается максимальное количество целевого продукта. Для решения этой задачи необходимо проанализировать влияние всех условий на протекание ХР. Решение можно осуществить несколькими путями — моделированием механизма и кинетики самой ХР [2], а также применением комплекса алгоритмов Data Mining к экспериментальным

данным о протекании ХР. Второй подход чаще всего является менее затратным с вычислительной точки зрения, однако требует существенного объема экспериментальных данных и разработки подходов к кодированию информации о факторах протекания химических реакций [1].

## 1. Формирование БД

Важным этапом второго подхода является формирование БД с результатами экспериментов при различных условиях. Данные об условиях протекания химических реакций являются слабосвязанными данными. В публикациях рассматривают влияние широкого набора факторов, данные могут быть указаны в разных единицах измерения. В рамках данной работы исследуется процесс протекания реакции карбонилирования олефинов. Данная ХР хорошо экспериментально изучена, поэтому удалось собрать достаточно представительную выборку, содержащую данные о влиянии различных условий на получение целевого продукта. В состав результирующей выборки включены:

- конверсия исходного вещества и селективность (для формирования выходных данных — целевые параметры),
- давление, температура, состав исходного вещества и состав катализатора (для входных параметров).

Данные о давлении были пересчитаны в атмосферах, данные о температуре приведены к градусам Цельсия, а на основе информации о химическом составе были сформированы атрибуты булева типа:

- l_has_p — в состав лиганда входит фосфор,
- substr_n_c — количество атомов углерода в исходном веществе,
- substr_ol — исходное вещество является спиртом,
- prec_cl — катализатор содержит хлор,
- acid_type — тип кислоты (органическая или не органическая).

Созданная выборка, содержащая 183 записи, является частью БД, содержащей больше входных параметров. Мы оставили наиболее информативные параметры, определенные по данным предварительных исследований.

## 2. Комплексный анализ данных

Эксперт, занимающийся анализом данных, обычно, хочет получить ответы на вопросы: «Каковы ожидаемые значения целевых параметров при заданном наборе входных параметров, каковы наиболее информативные параметры, влияющие на результат, и как управлять ими для достижения нужного выхода»? Зачастую для эксперта важны не столько конкретные значения параметров, сколько качественные оценки. Ему необходимы правила, позволяющие предсказать класс, которому принадлежит целевой параметр при управлении входными данными. Понятно, что эксперту нужны не только правила, но и характеристики этих правил, такие как достоверность правила, частота применения правила, возможно и другие характеристики. По нашему глубокому убеждению, ответы на эти вопросы можно получить, применяя широкий арсенал методов извлечения знаний из данных, известных как методы Data Mining.

## 3. Программная платформа

Для анализа данных мы используем программный комплекс, содержащий три модуля:

- Предварительная подготовка данных.
- Комплекс алгоритмов анализа данных.
- Визуализация и объяснение результатов анализа.

Для части алгоритмов анализа данных разработана собственная реализация, для части используются алгоритмы, входящие в пакет sklearn языка Python. Предварительная подготовка данных является важной частью работы с данными. На этом этапе решаются такие проблемы как восстановление пропусков в записях базы данных, а, главное, преобразование данных к виду, требуемому тем

или иным алгоритмом. Одни алгоритмы могут работать только с транзакционными данными, другие — с категориальными данными, третьи — с непрерывными данными, возможно, приведенными к одному масштабу. Комплекс алгоритмов включает алгоритмы:

- Построения ассоциативных правил.
- Различные вариации алгоритма кластеризации kmeans для работы с непрерывными данными и алгоритма CLOPE для работы с категориальными (транзакционными) данными.
- Построения деревьев решений для задачи классификации.
- Построения деревьев решений для задачи регрессии.

Остановимся на некоторых алгоритмах и результатах их работы.

## 4. Алгоритм ConApriori построения ассоциативных правил

У разработанного нами алгоритма ConApriori есть два существенных отличия в сравнении с классическим алгоритмом Apriori:

- способ представления данных,
- способ построения достоверных ассоциативных правил.

Каждая запись базы данных представлена одним числом. Числовое представление записей базы данных не зависит от размера записи и позволяет эффективно вычислять значение базовой функции Support за время $O(N)$ с минимальной константой, где $N$ — число записей базы данных. В отличие от классического алгоритма, в котором строятся частые правила на основе ранее построенных частых правил, в данном алгоритме строятся достоверные правила на основе ранее построенных достоверных правил, что повышает эффективность алгоритма. Подробное описание алгоритма приведено в работе [3]. Алгоритм позволяет находить правила с заданной частотой и достоверностью. Кроме этого, для каждого правила вычисляется характеристика, называемая lift, определяющая степень корреляции между посылкой правила и его заключением. Приведем

несколько правил, найденных в результате работы алгоритма для исследуемой БД:

$P2 \Rightarrow sel\_tar1$ : частота = 0,21;
достоверность = 0,93; лифт = 2,54

$P2, T1 \Rightarrow sel\_tar1$ : частота = 0,21;
достоверность = 0,93; лифт = 2,54

$P2, acid\_type3 \Rightarrow sel\_tar1$ : частота = 0,21;
достоверность = 0,95; лифт = 2,60

## 5. Алгоритм DecisionTreeClassifier построения дерева классификации

Этот алгоритм, реализованный в пакете tree, входящем в состав пакета sclearn, для построения дерева классификации использует два критерия — примесь Джини и энтропию. Примесь Джини считается по следующей формуле:

$$gini = \sum_{i=1}^{N}(1 - p_i^2) \tag{1}$$

Для критерия энтропии применяется известная формула Шеннона:

$$entropy = -\sum_{i=1}^{N} p_i * \log_2(p_i) \tag{2}$$

В обеих формулах используются $p_i$ — вероятности появления классов, рассчитываемые как частоты появления класса в выборке. Для наших данных оба критерия строят качественно похожие деревья. Приведем правила, которые можно вывести из анализа дерева решений. Дерево решений для классификации целевого параметра sel_tar строится по обучающей выборке, содержащей 128 записей, в которых целевой параметр представлен тремя классами примерно равной мощности. В корне дерева имеет место следующая ситуация: $(gini = 0.66, samples128(48, 33, 47))$. При построении дерева находятся два наиболее информативных параметра — давление и температура. При выполнении условий: $if(P > 52.2 \ \& \ T < 105)$ уже на 3-м

шаге приходим в узел ($gini = 0.37, samples34(1, 7, 26)$), где уже можно применять правило с примерной частотой 0.25 и достоверностью 0.75, о том, что целевой параметр sel_tar принадлежит третьему классу. При выполнении условия: $if(P >= 52.2 \quad \& \quad T < 72.5)$ приходим в узел (узел ($gini = 0.24, samples37(32, 1, 4)$). Здесь решение о том, что целевой параметр принадлежит к первому классу принимается примерно с той же частотой, но с более высокой достоверностью, приближающейся к 0.9. Эти данные хорошо согласуются с приведенными выше ассоциативными правилами.

Размер тезисов не позволяет даже кратко характеризовать другие используемые алгоритмы анализа данных.

## Заключение

1) Рассматриваемая нами БД позволяет извлечь знания, которые могут помочь эксперту, занимающемуся исследованием эффективности протекания процесса карбонилирования олефинов.
2) Комплексное применение алгоритмов Data Mining повышает надежность и способствует улучшению качества анализа данных.

## Список литературы

[1] *Биллиг, В. А.* Информационная система обработки и хранения данных о кинетике химических реакций / В. А. Биллиг, Н. В. Звягинцев // Программные продукты и системы. — 2018. — Т. 31, №. 4. — С. 808–813.

[2] Исследование влияния природы лигандов на региоселективность реакции карбонилирования стирола в присутствии комплексов палладия (II) / Н. В. Звягинцев, О. Л. Елисеев, Л. Т. Кондратьев, А. Л. Лапидус // Доклады Академии наук. — 2010. — Т. 434, №. 2. — С. 189–195.

[3] *Billig, V. A.* Effective algorithm for constructing associative rules // Программные продукты и системы. — 2017. — Т. 30, №. 2. — С. 196–206.

**Библиографическая ссылка**

*Биллиг, В. А.* Комплекс алгоритмов Data Mining в исследовании процесса протекания химических реакций / В. А. Биллиг, Н. В. Звягинцев // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 118–124.
https://doi.org/10.26456/mfcsics-21-19

**Сведения об авторах**

1. Владимир Арнольдович Биллиг
Тверской государственный технический университет. Профессор

*Россия, Тверь, наб. Афанасия Никитина, 22*
*E-mail: vladimir-billig@yandex.ru*

2. Николай Васильевич Звягинцев
Тверской государственный технический университет. Аспирант

*Россия, Тверь, наб. Афанасия Никитина, 22*
*E-mail: n.zvyagintsev@gmail.com*

УДК 519.7
AMS MSC2020: 68Q85

# Сравнение языков моделей сетей Петри со слабой временной стратегией

**Вирбицкайте И. Б., Зубарев А. Ю.**

Институт систем информатики им. А. П. Ершова СО РАН

Аннотация. Непрерывно-временные сети Петри (НВСП), где каждому переходу сети ставится в соответствие временной интервал его срабатывания, используются для моделирования сложных параллельных систем, критичных с точки зрения безопасности. В статье вводятся и исследуются языковые эквивалентности в семантиках интерливинга (одиночные действия), шага (множества параллельных действий) и частичного порядка (множества упорядоченных по причине и параллельных действий) в контексте НВСП, количественное (временное) поведение которых определяется слабой временной стратегией (то есть ход модельного времени не ограничен срабатыванием переходов сети), а качественное (функциональное) поведение представляется интерливинговыми/шаговыми пробегами и причинно-следственными процессами.

Ключевые слова: непрерывно-временные сети Петри, языки моделей, дихотомия «интерливинг — частичный порядок».

## Введение

Для верификации поведения информационно-компьютерных систем, безопасность функционирования которых критически важна, используются модели непрерывно-временных сетей Петри (НВСП), позволяющие описывать и анализировать как функциональные (качественные), так и реально-временные (количественные) свойства систем.

Классическая интерливинговая семантика НВСП представляется в виде пробегов — последовательностей смены состояний сети посредством хода времени и срабатываний одиночных переходов. При шаговой семантике смена состояний в пробеге НВСП осуществляется

посредством одновременного срабатывания множества параллельных переходов (шага). Для построения частично-упорядоченной семантики НВСП используют понятие временных процессов, состоящих из ациклических конструкций — временных причинно-следственных сетей (ВПСС), построенных из элементов, связанных частичным порядком (отношением причины), а отсутствие порядка соответствует параллелизму, — и отображений (гомоморфизмов) из ВПСС в НВСП. При формальной верификации моделируемых систем шаговая и частично-упорядоченная семантики НВСП позволяют сократить число анализируемых состояний, поскольку не требуется рассмотрение всех интерливинговых пробегов.

Различают две стратегии хода времени в НВСП. При *сильной стратегии* запрещен ход времени, приводящий к выходу за границы временных интервалов разрешенных переходов, которые должны срочно сработать, если время не может идти дальше. Напротив, при *слабой стратегии* допускается любой ход времени, срабатывание переходов не форсируется и, как следствие, они, вообще, могут не сработать. В [1] авторами было доказано, что эти две семантики являются несравнимыми относительно слабой временной бисимуляции. Кроме того, из работы [2] известно, что многие проблемы анализа НВСП разрешимы для слабой стратегии, однако это не всегда так для сильной стратегии.

Поведенческие эквивалентности и их взаимосвязи в дихотомиях «интерливинг — частичный порядок» и «линейное — ветвящееся время» в контексте НВСП с сильной стратегией были изучены в статье [3]. В данной работе вводятся и изучаются языковые эквивалентности в семантиках интерливинга, шага и частичного порядка для НВСП со слабой временной стратегией.

## 1. НВСП: интерливинговая/шаговая семантика

Определение 1. *Непрерывно-временная сеть Петри (НВСП) — это набор* $\mathcal{TN} = ((P, T, F, M_0, L), D)$, *где* $(P, T, F, M_0, L)$ — *сеть Петри с конечным множеством* $P$ *мест, конечным множеством* $T$ *переходов* ($P \cap T = \varnothing$), *отношением инцидентности* $F \subseteq (P \times T) \cup (T \times P)$, *начальной разметкой* $\varnothing \neq M_0 \subseteq P$ *и помечающей функцией* $L : T \to Act$; $D : T \to \{[a, b], [a, b) \mid a \in \mathbb{Q}_{\geqslant 0}, b \in (\mathbb{Q}_{\geqslant 0} \cup \{\infty\}), a \leqslant b\}$ —

*статическая временная функция. Здесь Act — множество действий.*

Для $x \in P \cup T$ и $X \subseteq P \cup T$ введем обозначения: ${}^\bullet x = \{y \mid (y,x) \in F\}$, $x^\bullet = \{y \mid (x,y) \in F\}$, ${}^\bullet X = \bigcup\limits_{x \in X} {}^\bullet x$, $X^\bullet = \bigcup\limits_{x \in X} x^\bullet$. Будем считать, что ${}^\bullet t \neq \varnothing$ и $t^\bullet \neq \varnothing$ для всех $t \in T$.

Разметка $M$ НВСП $\mathcal{TN}$ — это любое подмножество множества $P$. Обозначим через $En(M)$ множество переходов $t$ таких, что ${}^\bullet t \subseteq M$. Непустое подмножество $U \subseteq T$ называется шагом, если $({}^\bullet t \cup t^\bullet) \cap ({}^\bullet t' \cup t'^\bullet) = \varnothing$ для всех $t \neq t' \in U$. Шаг $U$ разрешен в разметке $M$, если $U \subseteq En(M)$. Состояние НВСП $\mathcal{TN}$ — это пара $S = (M, I)$, где $M$ — разметка и $I : En(M) \to \mathbb{R}_{\geqslant 0}$ — динамическая временная функция. Пара $S_0 = (M_0, I_0 \equiv 0)$ — начальное состояние $\mathcal{TN}$. Шаг $U$, разрешенный в разметке $M$, может сработать в состоянии $S = (M, I)$ (обозначается $U \in Fi(S)$), если $I(t) \in D(t)$ для всех $t \in U$.

Ход времени $\tau \in \mathbb{R}_{\geqslant 0}$ в состоянии $S = (M, I)$ приводит к новому состоянию $S' = (M, I')$ (обозначается $S \xrightarrow{\tau} S'$), где $I'(t) = I(t) + \tau$ для всех $t \in En(M)$. Срабатывание шага $U \in Fi(S)$ приводит к новому состоянию $S' = (M', I')$ (обозначается $S \xrightarrow{U} S'$) такому, что верно:

$$M' = (M \setminus {}^\bullet U) \cup U^\bullet;$$

$$\forall t \in En(M') : I'(t) = \begin{cases} 0, & \text{если } t \notin En(M \setminus {}^\bullet U) \vee t \in U, \\ I(t), & \text{иначе.} \end{cases}$$

Пусть $S_0 \xrightarrow{\theta_0} S'_0 \xrightarrow{U_1} S_1 \ldots S'_{n-1} \xrightarrow{U_n} S_n \xrightarrow{\theta_n} S'_n = S$ $(n \geqslant 0)$, $\theta_i \in \mathbb{R}_{\geqslant 0}$ $(0 \leqslant i \leqslant n)$ и $U_j \in 2^T$ $(0 < j \leqslant n)$. Тогда $\sigma = \theta_0\ U_1\ \theta_1\ \ldots\ U_n\ \theta_n$ — пробег НВСП $\mathcal{TN}$ (обозначается $\sigma \in \mathcal{FS}(\mathcal{TN})$), а $S$ — достижимое состояние в $\mathcal{TN}$. Будем считать, что для любого достижимого состояния $S = (M, I)$ и любого шага $U \in Fi(S)$ верно, что $(M \setminus {}^\bullet U) \cap U^\bullet = \varnothing$. Определим шаговый (интерливинговый) язык НВСП $\mathcal{TN}$ следующим образом:

$$\mathcal{L}_s(\mathcal{TN}) = \Big\{\theta_0 A_1 \theta_1 \ldots A_n \theta_n \mid \theta_0 U_1 \theta_1 \ldots U_n \theta_n \in \mathcal{FS}(\mathcal{TN}), \\ A_j = L(U_j) = \sum_{t \in U_j} L(t)\ (0 < j \leqslant n)\Big\}$$

$(\mathcal{L}_i(\mathcal{TN}) = \{\theta_0 A_1 \theta_1 \dots A_n \theta_n \in \mathcal{L}_s(\mathcal{TN}) \mid |A_j| = 1 \ (0 < j \leqslant n)\}).$

## 2. НВСП: частично-упорядоченная семантика

Определение 2. *Причинно-следственная сеть (ПСС) — это конечная ациклична сеть $N = (B, E, G, l)$, где $B$ — множество условий; $E$ — множество событий; $G \subseteq (B \times E) \cup (E \times B)$ — отношение инцидентности такое, что $|\, b^\bullet \,| \leqslant 1$ и $|\, {}^\bullet b \,| \leqslant 1$ для всех $b \in B$ и $E = {}^\bullet B = B^\bullet$; $l : E \to Act$ — помечающая функция.*

Транзитивное замыкание отношения $G$ (частичный порядок) определяет отношение причины ($\preceq$) на событиях и условиях сети. Отсутствие отношения причины между элементами сети говорит о их параллелизме. Непустое подмножество параллельных событий называется шагом; максимальное по включению подмножество параллельных условий называется сечением. Пусть $\mathcal{C}(N)$ — множество всех сечений в $N$. Для ПСС $N = (B, E, G, l)$ и $C, C' \in \mathcal{C}(N)$ введем обозначения:

$$\begin{aligned} {}^\bullet N &= \{b \in B \mid {}^\bullet b = \varnothing\}; \\ N^\bullet &= \{b \in B \mid b^\bullet = \varnothing\}; \\ \downarrow C &= \{e \in E \mid e \preceq e' \in {}^\bullet C\}; \\ C \smile C' &\iff \downarrow C \not\subseteq \downarrow C' \wedge \downarrow C' \not\subseteq \downarrow C. \end{aligned}$$

Определение 3. *Временная ПСС (ВПСС) — пара $TN = (N, \tau)$, где $N$ — ПСС и $\tau : \mathcal{C}(N) \to \mathbb{R}_{\geqslant 0} \cup \{\perp\}$ — временная функция такая, что для всех $C \in \mathcal{C}(N)$ верно:*

$$\tau(C) = \perp \iff \exists C' \smile C \colon \tau(C') > 0.$$

*Пусть $\mathcal{RC}(TN) = \{C \in \mathcal{C}(N) \mid \tau(C) \in \mathbb{R}_{\geqslant 0}\}$.*

Событие $e$ ВПСС $TN$ может произойти в $C \in \mathcal{RC}(TN)$ (обозначается $e \in Fi(C)$), если ${}^\bullet e \subseteq C$ и $((C \setminus {}^\bullet e) \cup e^\bullet) \in \mathcal{RC}(TN)$. Последовательность $\omega = C_0 \dots C_n$ $(n \geqslant 0)$ срезов из $\mathcal{RC}(TN)$ — график ВПСС $TN$, если $C_0 = {}^\bullet N$, $C_n = N^\bullet$ и $C_i = (C_{i-1} \setminus {}^\bullet V_i) \cup V_i^\bullet$, где $V_i \subseteq Fi(C_{i-1})$ — шаг, для всех $0 < i \leqslant n$. Для сечения $C_j$ $(0 \leqslant j \leqslant n)$ и события $e \in Fi(C_j)$ определим функцию $\mathbf{Clock}(C_j, e) = \sum_{\{C_k \mid {}^\bullet e \subseteq C_k, 0 \leqslant k \leqslant j\}} \tau(C_k)$. Заметим, что функция определена для каждого среза из $\mathcal{RC}(TN)$, а ее значение не зависит от выбора графика.

Определение 4. *Пусть* $\mathcal{TN} = ((P, T, F, M_0, L), D)$ — *НВСП и* $TN = (N = (B, E, G, l), \tau)$ — *ВПСС. Гомоморфизмом из* $TN$ *в* $\mathcal{TN}$ *называется отображение* $\varphi : (B \cup E) \to (P \cup T)$ *такое, что* $\varphi(B) \subseteq P$ *и* $\varphi(E) \subseteq T$; $\varphi|_{^\bullet e}$ — *биекция между* $^\bullet e$ *и* $^\bullet\varphi(e)$ *для всех* $e \in E$; $\varphi|_{e^\bullet}$ — *биекция между* $e^\bullet$ *и* $\varphi(e)^\bullet$ *для всех* $e \in E$; $\varphi|_{^\bullet N}$ — *биекция между* $^\bullet N$ *и* $M_0$; $l(e) = L(\varphi(e))$ *для всех* $e \in E$. *Пара* $\pi = (TN, \varphi)$ *называется временным процессом НВСП* $\mathcal{TN}$ *(обозначается* $\pi \in Proc(\mathcal{TN})$*), если для каждого сечения* $C \in \mathcal{RC}(TN)$ *и события* $e \in Fi(C)$ *верно, что* $\mathbf{Clock}(C, e) \in D(\varphi(e))$.

Временной процесс $\pi_0 = (TN_0 = (N_0 = (B_0, \varnothing, \varnothing, \varnothing), \tau_0), \varphi_0)$, где $\tau_0(^\bullet N_0 = B_0) = 0$ называется начальным. Рассмотрим $\pi$, $\widehat{\pi} \in Proc(\mathcal{TN})$. Будем писать $\pi \xrightarrow{A} \widehat{\pi}$, если существует шаг $V$ такой, что $\widehat{N}^\bullet = (N^\bullet \setminus {}^\bullet V) \cup V^\bullet$, $\widehat{B} = B \cup \widehat{N}^\bullet$, $\widehat{E} = E \cup V$, $\widehat{G} \cap (B \times E \cup E \times B) = G$, $\widehat{l}|_E = l$, $\widehat{\tau}|_{\mathcal{C}(N)} = \tau$, $\widehat{\varphi}|_E = \varphi$, $\sum_{e \in V} \widehat{l}(e) = A$ и $\widehat{\tau}|_{\mathcal{C}(\widehat{N}) \setminus \mathcal{C}(N)} \equiv 0$. Кроме того, будем писать $\pi \xrightarrow{\theta} \widehat{\pi}$, если $N = \widehat{N}$, $\tau|_{\mathcal{C}(N) \setminus N^\bullet} \equiv \widehat{\tau}|_{\mathcal{C}(N) \setminus N^\bullet}$, $\tau(N^\bullet) = 0$ и $\widehat{\tau}(N^\bullet) = \theta$.

## 3. Иерархия языковых эквивалентностей

Сначала рассмотрим языковые эквивалентности НВСП, построенные на их пробегах.

Определение 5. *НВСП* $\mathcal{TN}$ *и* $\mathcal{TN}'$ *являются шагово (интерливингово) языково эквивалентными (обозначается* $\mathcal{TN} \cong_{s(i)} \mathcal{TN}'$*), если* $\mathcal{L}_{s(i)}(\mathcal{TN}) = \mathcal{L}_{s(i)}(\mathcal{TN}')$.

Для НВСП $\mathcal{TN}$ введем вспомогательные обозначения:

$$Trace_s(\mathcal{TN}) = \{\theta_0 A_1 \theta_1 \dots A_n \theta_n \mid \theta_i \in \mathbb{R}_{\geqslant 0}\ (0 \leqslant i \leqslant n), A_j \in \mathbb{N}^{Act}\ (0 < j \leqslant n), \pi_0 \xrightarrow{\theta_0} \pi'_0 \xrightarrow{A_1} \pi_1 \dots \pi'_{n-1} \xrightarrow{A_n} \pi_n \xrightarrow{\theta_n} \pi'_n\ (n \geqslant 0)\ \text{в}\ \mathcal{TN}\};$$

$$Trace_i(\mathcal{TN}) = \{\theta_0 A_1 \theta_1 \dots A_n \theta_n \in Trace_s(\mathcal{TN}) \mid |A_j| = 1 (0 < j \leqslant n)\};$$

$$Trace_{pr}(\mathcal{TN}) = \{[TN]_\simeq \mid \pi = (TN, \varphi) \in Proc(\mathcal{TN})\}.$$

Определим языковые эквивалентности на временных процессах.

Определение 6. *Пусть* $\star \in \{i, s, pr\}$. *НВСП* $\mathcal{TN}$ *и* $\mathcal{TN}'$ *являются* $\star$*-языково эквивалентными (обозначается* $\mathcal{TN} \equiv_\star \mathcal{TN}'$*), если* $Trace_\star(\mathcal{TN}) = Trace_\star(\mathcal{TN}')$.

Теорема 1. *Пусть* $\mathcal{TN}$ *и* $\mathcal{TN}'$ *НВСП.*

$$\mathcal{TN} \cong_i \mathcal{TN}' \iff \mathcal{TN} \equiv_i \mathcal{TN}';$$
$$\mathcal{TN} \cong_s \mathcal{TN}' \iff \mathcal{TN} \equiv_s \mathcal{TN}'.$$

Установим взаимосвязи между языковыми процессными эквивалентностями НВСП.

Теорема 2. *Для двух НВСП* $\mathcal{TN}$ *и* $\mathcal{TN}'$ *выполняется:*

$$\mathcal{TN} \equiv_s \mathcal{TN}' \Rightarrow \mathcal{TN} \equiv_i \mathcal{TN}';$$
$$\mathcal{TN} \equiv_{pr} \mathcal{TN}' \Rightarrow \mathcal{TN} \equiv_s \mathcal{TN}';$$
$$\mathcal{TN} \equiv_i \mathcal{TN}' \not\Rightarrow \mathcal{TN} \equiv_s \mathcal{TN}';$$
$$\mathcal{TN} \equiv_s \mathcal{TN}' \not\Rightarrow \mathcal{TN} \equiv_{pr} \mathcal{TN}'.$$

## Заключение

В качестве дальнейшей работы планируется использовать разработанные в этой статье семантики для определения и изучения поведенческих эквивалентностей НВСП со слабой временной и различными пространственными стратегиями в дихотомии «линейное — ветвящееся время».

## Список литературы

[1] *Boyer, M.* Comparison of the expressiveness of arc, place and transition time Petri nets / M. Boyer, O. H. Roux // International Conference on Application and Theory of Petri Nets / Eds. J. Kleijn, A. Yakovlev. — Berlin, Heidelberg : Springer, 2007. — P. 63–82.

[2] *Reynier, P. A.* Weak time Petri nets strike back! / P. A. Reynier, A. Sangnier // International Conference on Concurrency Theory / Eds. M. Bravetti, G. Zavattaro. — Berlin, Heidelberg : Springer, 2009. — P. 557–571.

[3] *Virbitskaite, I.* True concurrent equivalences in time Petri nets / I. Virbitskaite, D. Bushin, E. Best // Fundamenta Informaticae. — 2016. — Vol. 149, №4. — P. 401–418.

### Библиографическая ссылка

*Вирбицкайте, И. Б.* Сравнение языков моделей сетей Петри со слабой временной стратегией / И. Б. Вирбицкайте, А. Ю. Зубарев // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 125–131.
https://doi.org/10.26456/mfcsics-21-20

### Сведения об авторах

1. Вирбицкайте Ирина Бонавентуровна
Институт систем информатики им. А. П. Ершова СО РАН. Главный научный сотрудник

*Россия, 630090, Новосибирск, пр. Академика Лаврентьева, д. 6*
*E-mail: virb@iis.nsk.su*

2. Зубарев Алексей Юрьевич
Институт систем информатики им. А. П. Ершова СО РАН. Аспирант

*Россия, 630090, Новосибирск, пр. Академика Лаврентьева, д. 6*
*E-mail: auzubarev@gmail.com*

УДК 510.644
AMS MSC2020: 03B22

# Субституциональные логики[1]

**Горбунов И. А.**

Тверской государственный университет

Аннотация. Рассматривается вопрос о существовании субституциональных логик, отличных от классической логики. Доказано, что любая табличная логика, имеющая функционально полную систему связок, является субституциональной. Для этих логик доказано существование алгоритма, который по рекурсивной непротиворечивой аксиоматике теории строит для нее точную унифицирующую подстановку. Приведен эскиз алгоритма, который каждой конечно аксиоматизируемой теории табличной логики с полной системой связок ставит в соответствие точную унифицирующую подстановку. Рассмотрен вопрос существования несубституциональных логик, в частности, доказано, что интуиционистская логика является несубституциональной.

Ключевые слова: обращение подстановки, точная унифицирующая подстановка, субституциональные логики, табличные логики, интуиционистская логика.

## Введение

Алфавитом языка пропозициональной логики будем называть набор, состоящий из счетного множества пропозициональных переменных $\Pi = \{p_i : i \geqslant 1\}$, не более чем счетного множества $\Sigma$ конечноместных функциональных символов (которые мы будем называть символами логических связок) и множества вспомогательных символов $\Upsilon = \{(, ), ,\}$. Множество всех формул в этом языке обозначим посредством $\Phi$.

Подстановкой будем называть гомоморфное продолжение отображения $\varepsilon : \Pi \to \Phi$ на множество всех формул. Поскольку это

[1]Работа выполнена при финансовой поддержке Российского научного фонда, проект 21-18-00195.

продолжение единственно, то обозначать его будем тоже $\varepsilon$. Для любого множества формул $\Gamma$ посредством $\varepsilon\Gamma$ будем обозначать результат применения подстановки $\varepsilon$ ко всем формулам этого множества. Посредством $\mathbf{E}$ обозначим множество всех подстановок.

Л о г и к о й будем называть соответствие $\vdash\ \subseteq 2^{\Phi} \times \Phi$, которое удовлетворяет условиям:

$$\begin{array}{ll}
\varphi \in \Gamma \Rightarrow \Gamma \vdash \varphi & \text{(рефлексивность);} \\
\Gamma \vdash \varphi \text{ и } \forall \psi \in \Gamma\ (\Delta \vdash \psi) \Rightarrow \Delta \vdash \varphi & \text{(транзитивность);} \\
\Gamma \vdash \varphi \Rightarrow \exists \Delta \subseteq \Gamma\ (\Delta \vdash \varphi \text{ и } \Delta \text{ — конечное}) & \text{(финитарность);} \\
\Gamma \vdash \varphi \Rightarrow \forall \varepsilon \in \mathbf{E}\ (\varepsilon(\Gamma) \vdash \varepsilon(\varphi)) & \text{(структурность).}
\end{array}$$

Т е о р и е й данной логики будем называть множество формул, замкнутое относительно каждой секвенции логики. Теорию $T$ будем называть н е п р о т и в о р е ч и в о й, если $T \neq \Phi$.

Если в логике есть секвенция вида $\varnothing \vdash \varphi$, то формулу $\varphi$ будем называть т а в т о л о г и е й логики.

О б р а щ е н и е м п о д с т а н о в к и $\varepsilon$ будем называть операцию взятия прообраза множества формул для отображения $\varepsilon$. Обозначать эту операцию будем $\varepsilon^{-1}$. Таким образом, для любого множества формул $\Gamma$ имеем: $\varepsilon^{-1}(\Gamma) = \{\varphi : \varepsilon\varphi \in \Gamma\}$.

Известен следующий факт, который получил название лемма Сушко ([6], стр. 14): для любой пропозициональной логики верно, что прообраз всякой ее теории при всякой подстановке также является теорией данной логики.

Множество, состоящее из всех прообразов множества всех тавтологий, взятых относительно каждой из подстановок, образует некоторое множество теорий. Естественным образом возникает вопрос о том, как связаны между собой множество всех непротиворечивых теорий и множество всех прообразов множества тавтологий. Известно [1], что в случае классической логики эти множества совпадают, поскольку для каждой непротиворечивой $T$ теории классической логики можно указать такую подстановку $\varepsilon_T$, при которой будет верно равенство $\varepsilon_T^{-1}(L) = T$, где $L$ — множество всех тождественно истинных формул. Такую подстановку будем называть т о ч н о й у н и ф и ц и р у ю щ е й п о д с т а н о в к о й д л я т е о р и и $T$, а логики, имеющие точные унифицирующие подстановки для каждой непротиворечивой теории, будем называть с у б с т и т у ц и о н а л ь н ы м и логиками.

В работе [1] поставлены открытые вопросы о том, существуют ли субституциональные логики, отличные от классической, и существуют ли несубституциональные логики.

## 1. Логики и матрицы

Пару $\mathbf{M} = \langle \mathbf{A}, D\rangle$, где $\mathbf{A}$ — алгебра на некотором непустом множестве $A$ и $D \subseteq A$, будем называть *логической матрицей*. Множество $D$ будем называть *выделенным множеством*. Полагаем, что операции алгебры $\mathbf{A}$ конечноместны.

Посредством $F^{\mathbf{A}}$ обозначим множество всех функций, которые можно образовать с помощью суперпозиции базовых функций алгебры $\mathbf{A}$ с использованием множества переменных $\Pi$. *Оценкой переменных на матрице* $\mathbf{M}$ будем называть отображение $\nu : \Pi \to A$. При некоторой оценке переменных $\nu$ всякая функция $f \in F^{\mathbf{A}}$ принимает некоторое значение, которое будем обозначать $\nu(f)$. Множество всех оценок переменных на матрице $\mathbf{M}$ обозначим $\Theta^{\mathbf{M}}$.

Матрицу $\mathbf{M}$ будем называть *подходящей матрицей для логики* $\mathbf{L}$, если всякой связке этой логики в алгебре $\mathbf{A}$ поставлена в соответствие операция той же местности. Таким образом, на подходящей матрице всякой формуле $\varphi$ логики $\mathbf{L}$ ставится в соответствие функция $f^{\varphi} \in F^{\mathbf{A}}$. *Значением (оценкой) формулы* $\varphi$ *при оценке переменных* $\nu$ называем значение функции $f^{\varphi}$ при оценке $\nu$. Таким образом, $\nu(\varphi) = \nu(f^{\varphi})$. Далее, говоря о матрицах некоторой логики $\mathbf{L}$, имеем ввиду только подходящие матрицы этой логики.

Для всякого множества формул $\Gamma$ посредством $\nu(\Gamma)$ будем обозначать множество всех значений, которые принимают формулы из этого множества при оценке $\nu$.

Будем говорить, что матрица $\mathbf{M} = \langle \mathbf{A}, D\rangle$ *характеризует* логику $\mathbf{L}$ (или *полна относительно логики* $\mathbf{L}$), если для любого $\Gamma \cup \{\varphi\} \subseteq \Phi$ выполняется условие

$$\forall \nu \in \Theta^{\mathbf{M}}\ (\nu(\Gamma) \subseteq D \Rightarrow \nu(\varphi) \in D) \Leftrightarrow \Gamma \vdash \varphi.$$

Такую матрицу будем называть *характеристической матрицей логики* и говорить, что она задает логику $\mathbf{L}$, а логику $\mathbf{L}$

будем называть логикой матрицы $\mathbf{M}$.[2]

Теорию $T$ логики $\mathbf{L}$ будем называть выполнимой на матрице $\mathbf{M} = \langle \mathbf{A}, D \rangle$, если существует такая оценка $\nu$, что $\nu(T) \subseteq D$. Из определения характеристической матрицы следует, что единственной невыполнимой теорией заданной ею логики является противоречивая теория Ф.

Логику будем называть табличной, если она характеризуется матрицей $\mathbf{M} = \langle \mathbf{A}, D \rangle$, в которой множество $A$ конечно.

## 2. Множество оценок

Поскольку множество переменных П линейно упорядоченно своими индексами, то каждая оценка переменных $\nu$ определяет последовательность $(\nu(p_i) : i \geqslant 1)$, которую будем обозначать тоже посредством $\nu$. Далее мы будем отождествлять оценку и эту последовательность.

Для всякой последовательности $\alpha$, длина которой не меньше $n$, посредством $[\alpha]_n$ будем обозначать начальный интервал этой последовательности длиною $n$.

Далее считаем, что матрица $\mathbf{M} = \langle \mathbf{A}, D \rangle$ задана, и множество всех оценок на этой матрице будем обозначать $\Theta$.

Пусть $\alpha \subseteq A^n$; коинициальным классом оценок слова (последовательности) $\alpha$ будем называть множество оценок $\alpha^* = \{\nu : [\nu]_n = \alpha\}$.

Рассмотрим множество теорий логики $\mathbf{L}$, которая задается матрицей $\mathbf{M}$.

Всякой непротиворечивой теории $T$ сопоставим множество всех оценок $\Theta_T$, при которых истинны все формулы данной теории, и множество $\Theta_T(n) = \{\alpha \in A^n : \alpha^* \cap \Theta_T \neq \varnothing\}$.

Заметим, что для всякой теории $T$ множество оценок $\Theta_T$ является компактным, то есть для него выполняется условие

$$\forall \nu \in \Theta(\nu \in \Theta_T \Leftrightarrow \forall i \geqslant 1(([\nu]_i)^* \cap \Theta_T \neq \varnothing)).$$

[2]Таким образом, множество тавтологий логики некоторой матрицы совпадает со множеством формул, истинных в этой матрице.

## 3. Граф оценок и граф теории

Пусть $\mathbf{L}$ — логика, имеющая непустое множество тавтологий $L$. Пусть, к тому же, $\mathbf{L}$ — табличная логика, которая задается матрицей $\mathbf{M} = \langle \mathbf{A}, D\rangle$.[3] Кроме того, считаем, что связки логики образуют функционально полную систему функций в алгебре $\mathbf{A}$. (Вопросы полноты функций $k$-значных логик изложены в [3], или более детально в [4], п. 10 и 12.) Поскольку множество $A$ — конечное, его элементы можно заиндексировать начальным отрезком множества натуральных чисел, поэтому будем считать, что $A = \{0, \ldots, k-1\}$ для некоторого $k \geqslant 2$.

Посредством $V$ обозначим множество всех непустых слов алфавита $\{0, \ldots, k-1\}$. Положим, что отношение $E \subseteq V^2$ — это такое отношение, что $(\alpha, \beta) \in E$ тогда и только тогда, когда слово $\beta$ продолжает слово $\alpha$ на один символ.

Для любой теории $T$ и для любого слова $\alpha \in V \cup \{\Lambda\}$ (здесь $\Lambda$ — пустое слово) определим множество $R(\alpha, T) \subseteq A$, которое будем называть множеством разрешенных теорией $T$ продолжений слова $\alpha$. Это множество мы определим следующим образом:

$$R(\alpha, T) = \{0 \leqslant a \leqslant k-1 : \exists \beta \in \Theta_T(h(\alpha)+1)([\alpha]_i = [\beta]_i, \beta_{h(\alpha)+1} = a)\}.$$

Заметим, что для любого слова $\alpha \in V \cup \{\Lambda\}$ и любой непротиворечивой тории $T$ верно, что $R(\alpha, T) \neq \varnothing$.

Рассмотрим граф $G = \langle V, E\rangle$. Индукцией по высоте вершины $\alpha \in V$ определим отображение $r : V \to \{0, \ldots, k-1\}$, которое будем называть раскраской графа $G$ в соответствии с теорией $T$.

Пусть $h(\alpha) = 1$; тогда

$$r(\alpha) = \begin{cases} \alpha_1, & \text{если } \alpha^* \cap \Theta_T \neq \varnothing, \\ a \in R(\Lambda, T), & \text{если } \alpha^* \cap \Theta_T = \varnothing. \end{cases}$$

Пусть для всех вершин, высота которых меньше $h(\alpha) = i$, раскраска определена; тогда

$$r(\alpha) = \begin{cases} \alpha_i, & \text{если } (r([\alpha]_1), \ldots, r([\alpha]_{i-1}), \alpha_i)^* \cap \Theta_T \neq \varnothing, \\ a \in R(\beta, T), & \text{если } (r([\alpha]_1), \ldots, r([\alpha]_{i-1}), \alpha_i)^* \cap \Theta_T = \varnothing, \end{cases}$$

[3] Следовательно, в матрице $\mathbf{M}$ множество $D$ не пусто.

где $\beta = (r([\alpha]_1), \ldots, r([\alpha]_{i-1}))$ Граф $G$, раскрашенный в соответствии с некоторой теорией $T$, будем обозначать $G_T$ и называть графом теории $T$.

## 4. Точная унифицирующая подстановка

Пусть $T$ — фиксированная непротиворечивая теория. По графу $G_T$ для каждого $n \geqslant 1$ определим функцию $f_n(p_1, \ldots, p_n)$ следующим образом: для любой вершины $\alpha$ высоты $n$ положим $f_n(\alpha) = r(\alpha)$. В силу полноты системы связок логики существует формула $\pi_n(p_1, \ldots, p_n)$, представляющая эту функцию. Таким образом, для любой оценки $\nu$ и любого $i \geqslant 1$ верно, что $\nu(\pi_i) = r([\nu]_i)$.

Подстановку $\varepsilon_T$ определим следующим образом: $\varepsilon_T p_i = \pi_i$ для любого $i \geqslant 1$.

Будет верна

Теорема 1. *Всякая непротиворечивая теория табличной логики с полной системой связок является прообразом множества всех тавтологий этой логики при некоторой подстановке.*

Функцию $\neg x = x + 1 \pmod k$ назовем циклическим отрицанием. Как известно, система функций $\{\neg x, \max(x, y)\}$ является полной для любой $k$-значной логики ([4], стр. 62), где $k \geqslant 2$. Таким образом, в силу Теоремы 1, всякая логика в языке со связками $\neg$ и max, которая задается $k$-значной матрицей (где $k \geqslant 2$), с выделенным множеством $D = \{k - 1\}$, является субституциональной.

Заметим, что из условий, которым удовлетворяет функция раскраски, и Теоремы 1 следует, что верна

Теорема 2. *Если для теории $T$ классической логики существует алгоритм, который по любой непустой конечной последовательности $\alpha \in V$ определяет, пусто ли множество $\alpha^* \cap \Theta_T$, то функция $\varepsilon_T$ вычислима.*

## 5. Алгоритм поиска точной унифицирующей подстановки для конечно аксиоматизируемых теорий

Пусть теория $T$ — конечно аксиоматизируемая теория $k$-значной логики. Не уменьшая общности, можно считать, что она аксиоматизируется конечным списком аксиом $\Gamma$ от переменных $p_1, \ldots, p_n$.

В силу конечности множества аксиом и конечности множества $A^n$, вопрос о принадлежности $\alpha \in A^n$ ко множеству $\Theta_T(n)$ рекурсивно разрешим. Таким образом, существует алгоритм, который ставит в соответствие этой теории точную унифицирующую подстановку.

В частности, можно предложить следующий алгоритм.

Для множества формул $\Gamma$ составим общую таблицу истинности $\nabla_\Gamma$.

Посредством $\nabla_\Gamma^D$ обозначим подтаблицу таблицы $\nabla_\Gamma$, состоящую из всех строк таблицы $\nabla_\Gamma$, в которых все формулы из $\Gamma$ принимают значения из выделенного множества. Для любой таблицы $\nabla$ посредством $S(\nabla)$ обозначим число строк в $\nabla$.

Положим $m = \lceil \log_k S(\nabla_\Gamma^D) \rceil$. По таблице $\nabla_\Gamma^D$ определим таблицу $\Delta_\Gamma$ следующим образом:

- если $S(\nabla_\Gamma^D) = k^m$, то $\Delta_\Gamma = \nabla_\Gamma^D$;
- если $S(\nabla_\Gamma^D) < k^m$, то $\Delta_\Gamma$ получим добавлением к таблице $\nabla_\Gamma^D$ ее последней строки $k^m - S(\nabla_\Gamma^D)$ раз.

Каждой переменной $p_i$ сопоставим функцию $f_i$, которая задается строкой ее значений, совпадающей со столбцом значений переменной $p_i$ в таблице $\Delta_\Gamma$.

Поскольку система связок языка, в котором мы рассматриваем логику, полна, то функции $f_i$ можно сопоставить некоторую выражающую ее формулу, которую мы будем обозначать $\pi_i$.

Рассмотрим подстановку $\varepsilon_\Gamma$, которую определим следующим образом:

$$\varepsilon_\Gamma p_i = \begin{cases} \pi_i, & \text{если } 1 \leqslant i \leqslant n, \\ p_i, & \text{если } i > n. \end{cases}$$

Несложно заметить, что эта подстановка является точной унифицирующей подстановкой для теории $T$.

## 6. Несубституциональные логики

Пусть логика $\mathbf{L}$ задана. Множество всех тавтологий этой логики обозначим посредством $L$. Множество всех теорий этой логики обозначим посредством $Th(\mathbf{L})$. Посредством $T(\Gamma)$ обозначим наименьшую теорию, содержащую множество формул $\Gamma$.

Как известно ([6], стр. 196, Теорема 3.1.5), всякая логика характеризуется множеством всех матриц Линденбаума, более того, матрица $\langle \Phi^*, L\rangle$ (где $\Phi^*$ — свободная алгебра формул) является характеристической для множества всех тавтологий логики **L**.

Теорема 3. *Всякая субституциональная логика имеет характеристическую матрицу.*

Доказательство. Пусть логика **L** субституциональна.

Пусть $\Gamma \nvdash \varphi$. Тогда на матрице $\langle \Phi^*, L\rangle$ при оценке $\varepsilon_{T(\Gamma)}$ мы получим, что $\varepsilon_{T(\Gamma)}(\Gamma) \subseteq L$ и $\varepsilon_{T(\Gamma)}(\varphi) \notin L$. □

Таким образом, логика Йоханссона (Johansson) $J_{min}$ ([6], стр. 134), в силу Теоремы 3.2.7 ([6], стр. 206), Теоремы 3.2.10 ([6], стр. 208) и Теоремы 3, субституциональной не является, так как не имеет характеристической матрицы.

Заметим, что утверждение, обратное Теореме 3, верным не является, поскольку верна

Теорема 4. *Интуиционистская логика не является субституциональной.*

Как известно, интуиционистская логика характеризуется бесконечной матрицей ([2], стр. 121 или [5], стр. 58).

## Заключение

Мы доказали, что всякая табличная логика с функционально полной системой связок является субституциональной. Также было доказано, что любая субституциональная логика имеет характеристическую матрицу. При этом наличие у логики характеристической матрицы не обеспечивает ее субституциональности.

В связи с полученными в работе результатами возникают некоторые открытые вопросы.

Существуют ли несубституциональные табличные логики?

Верна ли следующая гипотеза?

Гипотеза 1. *Всякая теория интуиционистской логики является прообразом, при некоторой подстановке, наибольшей содержащейся в ней подтеории, замкнутой по всем подстановкам.*

### Список литературы

[1] *Горбунов, И. А.* Обращение подстановки и теории классической пропозициональной логики // Сборник трудов конференции «Алгебра и математическая логика: теория и приложения». — Казань : Изд-во КФУ, 2019. — С. 98–100.

[2] *Драгалин, А. Г.* Математический интуиционизм введение в теорию доказательств. — М. : Наука, 1979. — 256 с.

[3] *Яблонский, С. В.* Введение в дискретную математику : учебник. — Изд. 4-е, стер. — М. : Высшая школа, 2003. — 384 с.

[4] *Яблонский, С. В.* Функциональные построения в $k$-значной логике // Сборник статей по математической логике и ее приложениям к некоторым вопросам кибернетики. Тр. МИАН СССР. — 1958. — Т. 51. — С. 5–142.

[5] *Chagrov, A.* Modal Logic / A. Chagrov, M. Zakharyaschev. — Oxford : Clarendon Press, 1997. — 605 с.

[6] *Wojcicki, R.* Lectures on Propositional Calculi: Basic Theory of Consequence Operations. — Wroclaw : Ossolineum, 1984. — 473 с.

### Библиографическая ссылка

*Горбунов, И. А.* Субституциональные логики // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 132–140.
https://doi.org/10.26456/mfcsics-21-21

### Сведения об авторах

Горбунов Игорь Анатольевич
Тверской государственный университет. Доцент

*Россия, 170100, Тверь, ул. Желябова, 33, ТвГУ*
*E-mail: i_gorbunov@mail.ru*

УДК 001.32, 004.43
AMS MSC2020: 01A60, 01A70

# От дискретной математики к семантике языков программирования

**Городняя Л. В.**

Институт систем информатики им. А. П. Ершова СО РАН

Аннотация. Статья посвящена малоизвестной истории небольшого коллектива математиков, сложившегося в Новосибирске вокруг Б. А. Трахтенброта. Этот коллектив осуществил для отечественного программирования крайне важную работу по ознакомлению с мировыми достижениями на стыке теории и практики, включая выполнение переводов основополагающих работ в сфере анализа и компиляции языков программирования и развития математических моделей, лежащих в основе доказательного программирования, верификации программ и логического вывода свойств программ.

Ключевые слова: дискретная математика, конструктивная математика, аксиоматическая теория множеств, синтаксическое управление, аналогии, анализ языков программирования, граница автоматизации процессов, клиент-серверные системы, языкотворчество.



Тверской государственный университет

*Памяти Михаила Иосифовича*
*Дехтяря посвящается*[1]

**Немного истории**

В начале 1970-х годов сотрудники лаборатории Б. А. Трахтенброта[2] из Института математики СО АН В. Н. Агафонов[3], М. К. Валиев[4], М. И. Дехтярь[5] и В. Ю. Сазонов[6] вели в стенах Вычислительного центра СО АН семинар, посвященный семантике языков программирования. Слушателями семинара были преимущественно сотрудники лабораторий А. П. Ершова и И. В. Поттосина, а также НФ ИТМиВТ[7]. На этом семинаре дотошно рассматривались самые новые зарубежные работы, переводы которых потом публиковались в самых лучших издательствах [1, 3]. Качество переводов было выше всяких похвал — они читались так, будто сразу были написаны на русском языке, без тени стилистического влияния англоязычного оригинала [3]. Неспешный стиль, характерный для семинаров и лекций у математиков, не ставил задачи втиснуть доклад в заранее оговоренный регламент. Каждая тема рассматривалась без временных ограничений до достижения ясности в понимании идей и механизмов, при необходимости с переносом доклада на следующее заседание. Следует отдельно отметить уважительное отношений к практическому программированию [11]. Одной из сквозных тем было сопоставление

[1] Михаил Иосифович Дехтярь — лучший выпускник Механико-математического факультета НГУ 1969 года, сокурсник автора.

[2] Борис Авраамович Трахтенброт (1921–2016) — мировой авторитет в области теории автоматов, принял у автора устный вступительный экзамен в НГУ по математике [32].

[3] Валерий Николаевич Агафонов дал автору ценный совет по унификации сигнатур семантических систем, к которым сводима семантика языков и систем программирования.

[4] Марс Котдусович Валиев показал удивительные эффекты, присущие динамическим логикам [5, 6].

[5] Михаил Иосифович Дехтярь создал дистанционные курсы по дискретной математике, получившие высокую оценку студентов ММФ НГУ.

[6] Владимир Юрьевич Сазонов познакомил слушателей семинара с элегантностью кумулятивной иерархии множеств, по факту неявно присущей структурам данных многих языков программирования [28].

[7] Новосибирский филиал Института точной механики и вычислительной техники АН СССР.

функционального и императивного программирования. В данной статье дан краткий обзор тематики семинара.

## 1. Базовые понятия

Тематическую основу семинара составляло изучение математических моделей, имеющих связь с программированием и в наши дни. Появилась книга Клини С. К. «Введение в метаматематику» [21], освещающая основы конструктивной математики Л. Брауэра, позволяющая рассуждать о больших объектах без риска наткнуться на парадокс. Это послужило поводом рассмотреть разные варианты аксиоматических теорий множеств и моделей вычислений, включая кумулятивную иерархию, комбинаторы и упрощенную теорию множеств [47]. В центре внимания было различие в понимании механизмов определения, принадлежит ли элемент множеству. Многие варианты позднее можно было видеть в электронном курсе по проверке доказательств теорем у Патрика Суппеса [54].

Не менее основательно анализировалось понятие «функция» как механизм сопоставления одних элементов другим, результатов функции ее аргументам, что можно рассматривать как планарную проекцию решения задачи в виде отображения аргументов в результаты. Такой механизм давал возможность перехода к размеченному множеству для оперирования номерами, именами, адресами и ассоциациями, тем самым поддерживая переход к третьему измерению в процессе программирования решений задач — организации структур данных. Важно, что аппарат отображений способен эффективно поддержать использование подобий, аналогий, синтаксического управления, конструирования обработчиков регулярных структур данных произвольной природы, при условии сохранения гомоиконности между обрабатываемыми данными и программами их обработки [17]. Все это теперь называют мета-программированием, начало которого восходит к идеям языков Lisp и Рефал [33, 37, 48].

Уместно процитировать слова Пойа об аналогии: «Возможно, не существует открытий ни в элементарной, ни в высшей математике, ни даже, пожалуй, в любой другой области, которые могли бы быть сделаны... без аналогии» [29].

Организация вычислений с помощью функций обычно использует определенный порядок вычисления аргументов с последующим

применением конкретной функции. Появляется место для упорядоченных элементов данных — последовательности, вектора, стеки, строки. Возникает общая модель формулы как упорядоченного множества символов для представления значений, функций и выражений над ними с помощью данных, устроенных как конструкции из символов. Это позволяет выполнять символьные подстановки для преобразования любых формул — символьные вычисления.

Конечно, обсуждались новые результаты Д. Скотта[8], предложившего модель решеток в дискретном пространстве, позволяющую распространить на них технику предельных переходов, похожую на стремление к пределам в непрерывной математике [50–53]. Тем самым завершилась активная критика идей Джона Маккарти, реализованных в языке Lisp [48]. Определенное внимание было уделено $\lambda$-исчислению А. Черча [41], теории комбинаторов Х. Карри [19], продукциям Э. Поста [36] и теории категорий в изложении С. Фефермана [44].

В этом пространстве понятие «процесс» выглядит как последовательность получения результатов применения функций к предварительно определенным данным, а данное — это представление значений, множеств, функций. И программа — данное, представляющее процесс вычислений, что можно рассматривать как свертку процесса управления вычислениями. Объединяя в одно множество данных представления аргументов и результатов с функциями и формулами, получаем более сложную высоко уровневую планарную проекцию программируемых решений задач и подзадач в виде именованного отображения аргументов в результаты, допускающие массовое применение мета-программирования и синтаксически ориентированных методов конструирования.

И снова вспомним Пойа: «Недостаточно лишь понять задачу, необходимо желание решить ее. Без сильного желания решить трудную задачу невозможно, но при наличии такового возможно. Где есть желание, найдется путь!» [29]

[8] Д. Скотт является почетным профессором ММФ НГУ.

## 2. Расширенные понятия

Тогда, как и сейчас, при оценке потенциала языков программирования традиционно большое значение придавали полноте по Тьюрингу [4, 34, 35]. Появилось немало работ, показывающих, что для такой полноты не требуется особо сложных средств. Так макрогенератор JPM, предложенный Стрейчи [4], обеспечивает полноту простым механизмом именованных определением функций над текстами в общей памяти. Определения новых языков программирования в те годы было принято начинать с объявления специальных расширенных форм Бэкуса – Наура, приспособленных к описанию именно определяемого языка и гарантирующего корректность и эффективность анализа текстов программ. Интересным было знакомство с языком Рефал, весьма своеобразно конструирующим обработчики синтаксических формул — начало мета-программирования [33]. По мотивам языка BLISS [55] сотрудниками НФ ИТМиВТ был создан язык системного программирования ЯРМО, успешно применявшийся в проектах разработки программного обеспечения МВК Эльбрус [8, 9]. Разработчики языка ЯРМО были активными участниками семинара.

На этом уровне основной объем научных обсуждений был сконцентрирован на грамматиках и методах анализа языков программирования с учетом проблем спецификации и верификация программ. В. Н. Агафонов лично побеседовал с участниками практически всех коллективов, занимающихся разработкой систем программирования, включая реализацию одного из самых сложных языков Algol-68 [27, 31], что позволило ему представить исчерпывающе полный обзор практичных решений, находок и изобретений в этой сфере.

Контекстно-свободный синтаксис подвергался нагрузке контекстно-зависимой семантикой разными методами [17, 22, 26]. Наиболее практичными оказались идеи нормализованных и канонических форм синтаксиса, с которыми связаны ассоциированные автоматы генерации и анализа текстов [46]. Они быстро обрели инструментальную поддержку программными системами LEX и YACC, теперь унаследованными комплексом Clang-LLVM. Особенности представления семантики наиболее ясно выражены в работах по абстрактным машинам и в Венской методике определения языков программирования, сводящей семантику к паре промежуточных языков уровня абстрактного синтаксиса и абстрактной машине [43].

Каждая синтаксическая позиция обычно связана с семейством подъязыков, включающих вхождение элемента из этой позиции, возможно требующего отдельного механизма. Цепочка из соседних понятий связана с пересечением этих подъязыков или с вхождениями подъязыка в общий язык. Полная цепочка текстов, принадлежащих языку, может сводиться к пределу таких пересечений и объединений, возможно пустому. Удобно, когда предел является одноэлементным, но это не может быть гарантировано.

Математики до сих пор нередко отмечают бестеоремность научных работ по программированию. Тем не менее, значительная «теоремность» складывалась в программировании независимо от коньюнктурных проблем исключительно для нужд решения внутренних задач системного программирования: в методах разработки информационных систем, в оптимизации программ и в линии доказательного программирования [20], затем в методах спецификации, верификации и анализа надежности сложных систем [25, 49], включая автоматическую проверку доказательств теорем. Последнее не обрело популярности у математиков, возможно по той причине, что обнаружилось много неточностей в доказательствах, рассматриваемых математиками как корректные.

Как-то ускользнуло от внимания в те годы, что вся работа системного программиста пронизана доказательными построениями, подобными доказательствам теорем в математике. Любая программная система основана на определенной скрытой интуитивной теории. При проектировании и описании системы ее разработчик обычно формулирует высказывания, имеющие ранг теорем и выстраивает доказательные построения, показывающие справедливость таких теорем. В отличие от математических теорем, такие высказывания не обладают общезначимостью и не представляют интереса для развития математики. Для каждой программной системы их необходимо доказывать заново, подобно тому, как теоремы классической математики пришлось отдельно доказывать в конструктивной математике. Если доказано утверждение, что «В системе «А» любое хранимое данное можно представить с помощью символьных строк и эти строки выводить на экран или в файл», то для системы «В» его надлежит доказывать заново. Таких «теорем» и их доказательств при разработке любой нетривиальной программной системы неявно формулируется и незаметно доказывается несколько десятков. Такая,

узко прикладная, чисто математическая работа, требует владения техникой доказательств, которая по недоразумению считается для образования программистов не очень важной.

Примерно в эти годы идет внедрение ЕС ЭВМ, ориентированное отчасти на разработку производственных автоматизированных систем управления (АСУ). Обучать первокурсников специфике баз данных для АСУ в 1970-е было слишком долго — результат через 5 лет. Было организовано 2-хгодичное обучение околокомпьютерного персонала с незаконченным высшим образованием и опытом работы на ЭВМ, по специальным программам, ориентированным на эксплуатацию ЕС и разработку АСУ.

Интересным оказалось ознакомление с языком СУБД SQL [10]. Следует отметить, что среди разработчиков систем программирования считалось аксиомой, что выиграть скорость вычислений можно лишь смирившись с потерями памяти и, наоборот, — экономия памяти влечет неизбежную потерю скорости. Оказалось, что в мире баз данных изобретены подходы, дающие выигрыш по этим направлениям одновременно. Принципиальная разница заключается в границе автоматизации процесса принятия решений. Для систем программирования принято, что решения по оптимизации программ принимает компилятор на основе статического анализ программ. Для СУБД характерно, что решения относительно доступа к данным принимает пользователь в терминах нормализованных форм, допускающих выигрыш в скорости и памяти одновременно.

## 3. Парадигмы программирования и новые языки программирования

По мере расширения сферы применения вычислительной техники проявились неожиданные проблемы и возникли новые языки программирования и анализируются популярные (APL, Pascal, Setl, Simula-67, Snobol, Planner, Ada), поддерживающие разные парадигмы программирования [2, 7, 12–14, 18, 23, 24, 27, 28, 33, 37, 48, 55]. Роль такого разнообразия выражена в Тьринговских лекциях Р. Флойда и Дж. Бэкуса [42, 45]. Становится ясным, что проблемы программирования и практичные методы их решения не вполне удобно вписываются в пространство моделей классической математики. Возникает широкий спектр исследований по взаимодействию программ с внешним

миром, анализу и формализации постановок задач, представлению методов их решения и спецификация требований к качеству реализации решений. Далее идет исследование методов производства программ, техники программирования, разработки, реализации и отладки программ решения разных задач с удостоверением достаточного соответствия между программой, требованиями и постановкой задачи. Активно исследуется системное программирование, включая внутренние задачи повышения производительности труда, создания и совершенствования инструментария, операционных систем, языков и систем программирования, средств поддержки программистских проектов и т. п. Несколько в стороне остается формирование пользовательской потребности в средствах, системах, информационных технологиях, расширяющих круг решаемых задач, решение которых требует профессионального программирования.

После отъезда Б. А. Трахтенброта в декабре 1980 г. его сотрудники покинули Новосибирск. Их деятельность в Москве и Твери обрела более производственный характер, преимущественно связанный с разработкой и применением баз данных. Сохраняя бесспорную теоремность ранее выполненных исследований, они продолжили совместную работу и на уровне публикаций, сдвинув центр внимания с семантики языков программирования на проблемы БД и мультиагентных систем. Теоремная линия переключилась на исследование проблем организации клиент-серверных систем в Интернете. Многие результаты в этом направлении констатируют математическую неразрешимость практических задач клиент-серверной обработки данных.

В 2004 году появился Интернет-Университет Информационных технологий, для которого М. И. Дехтярь создал дистанционные курсы по математике, лежащей в основах программирования [15, 16]. Эти курсы получили высокую оценку студентов ММФ НГУ. Краткий начальный курс «Введение в схемы, автоматы и алгоритмы» знакомит с такими дискретными структурами как схемы, конечные автоматы и алгоритмы. В аннотации[9] курса отмечено, что изучаются

[9] Аннотации приведены полностью сознательно, чтобы привлечь внимание к ресурсам Интернет-Университета Информационных Технологий, располагающим 123 курсами по программированию и программному обеспечению и 76 по математике: http://www.intuit.ru

«специальные классы ориентированных графов без циклов: логическими схемами (схемами из функциональных элементов) и упорядоченными бинарными диаграммами решений. Рассмотрены основы теории конечных автоматов: конечные автоматы-преобразователи и -распознаватели, детерминированные автоматы и языки, недетерминированные автоматы и их детерминизация, регулярные выражения и языки, синтез конечного автомата по регулярному выражению, замкнутость класса автоматных языков относительно разных операций, теорема о разрастании для автоматных языков, примеры неавтоматных языков. Дается краткое введение в теорию алгоритмов, сравниваются три формальных модели описания алгоритмов: структурированные программы, частично рекурсивные функции и машины Тьюринга, формулируется тезис Тьюринга – Черча и устанавливается алгоритмическая неразрешимость ряда проблем, относящихся к свойствам структурированных программ. Решение большинства рассматриваемых в курсе проблем доведено до уровня алгоритмических процедур и проиллюстрировано на примерах. Каждая лекция завершается разделом с задачами и упражнениями, позволяющими закрепить пройденный материал» [15].

Начальный курс «Основы дискретной математики» знакомит с математическими моделями и дискретными структурами, необходимыми для изучения основных методов программирования в терминах множествах, комбинаторики и математической индукции. В аннотации отмечено, что в курсе «рассмотрен самый простой и важный класс дискретных функций — булевы функции: их различные представления, связь с логикой высказываний, основные логические тождества («законы логики»), дизъюнктивные и конъюнктивные нормальные формы и многочлены Жегалкина, полные системы функций (теорема Поста), задача выводимости для Хорновских формул. Даны краткое введение в логику предикатов и устанавливаются связи между ней и реляционными базами данных, введение в теорию графов, включающее представления графов, граф достижимости, компоненты сильной связности и базы ориентированного графа, деревья, их обходы, связь деревьев и формул (выражений), три классические задачи теории графов: построение минимального остова, обход графа в глубину (задачу о лабиринте) и задачу о кратчайших путях. Решение большинства рассматриваемых в курсе проблем доведено до уровня алгоритмических процедур

и проиллюстрировано на примерах. Каждая лекция завершается разделом с задачами и упражнениями, позволяющими закрепить пройденный материал» [16].

### Наши дни

Яркий интеллектуальный взлет пионерской эпохи программирования несколько растворился на фоне стремительного расширения сфер применения информационно-компьютерных технологий и превращения программирования в массовую профессию. Не получили практичного решения образовательные проблемы программирования и вопросы удостоверения разумности программируемых решений. Преимущественные направления деятельности свелись к композиции кодирования, интерфейсов и готовых библиотечных модулей. Даже ведущие фирмы-изготовители компьютеров и новых чипов теперь испытывают трудности в поиске специалистов, способных к приаппаратному программированию и изобретению решений новых задач. Справедливости ради следует отметить, что последние десятилетия характеризуются интенсивным языкотворчеством в сфере проблемно ориентированных (DSL) языков программирования, что можно рассматривать как переход от обычного программирования к мета-программированию. Тем не менее, переход от моделей дискретной математики к практичным моделям семантики языков программирования до сих пор не завершен [38–40].

### Список литературы

[1] *Агафонов, В. Н.* Спецификация программ: понятийные средства и их организация. — Новосибирск : Наука, 1987. — 240 с.

[2] *Айлиф, Дж.* Принципы построения базовой машины. — М. : Мир, 1973. — 119 с.

[3] *Ахо А.* Теория синтаксического анализа, перевода и компиляции : в 2 томах / А. Ахо, Дж. Ульман ; пер. с англ. В. Н. Агафонова под ред. В. М. Курочкина. — М. : Мир, 1978. — 2 т. — URL: http://rema44.ru/resurs/study/compiler1/present/Comp1-L01.pdf

[4] *Браун, П.* Макропроцессоры и мобильность программного обеспечения. — М. : Мир, 1977. — 253 с.

[5] *Валиев, М. К.* О пропозициональных программных логиках // Вопросы кибернетики. Неклассические логики и их применение. — М. : Наука, 1982. — С. 23–36.

[6] *Валиев, М. К.* Организация параллельных вычислений на системе с локальными взаимодействиями элементов / М. К. Валиев, А. И. Мишин // Автометрия. — 1983. — №6. — С. 88–96.

[7] *Вегнер, П.* Программирование на языке Ада. — М. : Мир, 1983. — 239 с.

[8] *Гололобов В.И.* Описание языка ЯРМО. Машинно-независимое ядро / В. И. Гололобов, Б. Г. Чеблаков, Г. Д. Чинин. — Новосибирск, 1980. — (Препринт / ВЦ СО АН СССР; №247).

[9] *Гололобов В.И.* Описание языка ЯРМО. Макросредства / В. И. Гололобов, Б. Г. Чеблаков, Г. Д. Чинин. — Новосибирск, 1980. — (Препринт / ВЦ СО АН СССР; №248).

[10] *Грабер, М.* Введение в SQL. — М. : Лори, 1996. — 377 с.

[11] *Грис Д.* Наука программирования. — М. : Мир, 1984. — 416 с.

[12] *Грисуолд, Р.* Язык программирования Снобол-4 / Р. Грисуолд, Дж. Поудж, И. Полонски. — М. : Мир, 1980. — 268 с.

[13] *Грогоно, П.* Программирование на языке Паскаль. — М. : Мир, 1982. — 382 с.

[14] *Дал, У.* Симула-67 универсальный язык программирования / У. Дал, Б. Мюрхауг, К. Нюгорд. — М. : Мир, 1969. — 99 с.

[15] *Дехтярь, М. И.* Основы дискретной математики. — URL: https://intuit.ru/studies/courses/1084/192/info. — Загл. с титул. экрана.

[16] *Дехтярь, М. И.* Введение в схемы, автоматы и алгоритмы. — URL: https://intuit.ru/studies/courses/1030/205/info. — Загл. с титул. экрана.

[17] *Ингерман, П.* Синтаксически ориентированный транслятор. — М. : Мир, 1969. — 174 с.

[18] *Йодан, Э.* Структурное проектирование и конструирование программ. — М. : Мир, 1979. — 409 с.

[19] *Карри, Х. Б.* Основания математической логики ; пер. с англ. — М. : Мир, 1969. — 568 с.

[20] *Касьянов, В. Н.* Сборник заданий по практикуму на ЭВМ / В. Н. Касьянов, В. К. Сабельфель. — М. : Наука, 1986. — 271 с.

[21] *Клини, С. К.* Введение в метаматематику ; пер. с англ. — М. : Иностранная литература, 1957. — 526 с.

[22] *Лавров, С. С.* Методы задания семантики языков программирования // Программирование. — 1978. — №6. — С. 3–10.

[23] *Лавров, С. С.* Универсальный язык программирования. — М. : Наука, 1972. — 183 с.

[24] *Магариу, Н. А.* Язык программирования АПЛ. — М. : Радио и связь, 1983. — 96 с.

[25] *Непомнящий, В. А.* Надежность оборудования электрических сетей 220–750 кВ энергосистем / В. А. Непомнящий, Л. А. Дарьян. — Москва : Энергопрогресс : Энергетик, 2018. — 123 с.

[26] *Оллонгрен, А.* Определение языков программирования интерпретирующими автоматами. — М. : Мир, 1977. — 288 с.

[27] Пересмотренное сообщение об АЛГОЛЕ 68 / Под ред. А. П. Ершова. — М. : Мир, 1979. — 534 с.

[28] *Пильщиков, В. Н.* Язык Плэнер. — М. : Наука, 1983. — 207 с.

[29] *Пойа, Дж.* Математическое открытие. Решение задач: основные понятия, изучение и преподавание. — М. : Наука, 1976. — 448 с.

[30] *Сазонов, В. Ю.* Последовательно и параллельно вычислимые функционалы // Сибирский математический журнал. — 1976. — Т. 17, №. 3. — С. 648–672.

[31] *Терехов, А. Н.* Язык синтеза объектной программы с учетом последующего контекста /А. Н. Терехов, Г. С. Цейтин // Труды всесоюзного симпозиума по методам реализации новых алгоритмических языков. — Новосибирск, ВЦ СО АН СССР, 1975. — С. 227–236.

[32] *Трахтенброт, Б. А.* Алгоритмы и вычислительные автоматы. — М. : Советское радио, 1974. — 200 с.

[33] *Турчин, В. Ф.* РЕФАЛ-5. Руководство по программированию и справочник. — URL: http://refal.net/rf5_frm.htm. — Загл. с титул. экрана.

[34] *Тьюринг, А.* Может ли машина мыслить? (С приложением статьи Дж. фон Неймана Общая и логическая теория автоматов. Пер. и примечания Ю. В. Данилова). — М. : ГИФМЛ, 1960. — 68 с.

[35] *Тьюринг, А.* Теория игр и экономическое поведение / А. Тьюринг, Дж. Нейман, О. Моргенштерн. — М. : Наука, 1970. — 983 с.

[36] *Успенский, В. А.* Машина Поста. — 2-е изд., испр. — М. : Наука, 1988. — 96 с.

[37] Физики шутят / Сост.-пер. Ю. Конобеев, В. Павлинчук, Н. Работнов, В. Турчин ; под общей ред. В. Турчина. — М. : Мир, 1966. — 162 с.

[38] *Хигман, Б.* Сравнительное изучение ЯП. — М. : Мир, 1974. — 204 с.

[39] *Хоар, Ч.* Взаимодействующие последовательные процессы. — М. : Мир, 1989. — 264 с.

[40] *Цейтин, Г. С.* Ассоциативное исчисление с неразрешимой проблемой эквивалентности // Проблемы конструктивного направления в математике. Тр. МИАН СССР. — 1958. — Т. 52. — С. 172–189.

[41] *Черч, А.* Введение в математическую логику : в 2 томах. Т. 1. / Пер. с англ. В. С. Черняевского под ред. В. А. Успенского. — М. : Издательство иностранной литературы, 1960. — 484 с.

[42] *Backus, J.* Can programming be liberated from the von Neumann style? A functional stile and its algebra of programs // Communications of the ACM. — 1978. — Vol. 21, №8. — P. 613–641.

[43] The Vienna Development Method: The Meta-Language. LNCS'61 / Eds. D. Bjørner, C. B. Jones. — Berlin, Heidelberg : Springer-Verlag, 1978. — 384 p.

[44] *Feferman, S.* Categorical Foundations and Foundations of Category Theory // Logic, Foundations of Mathematics, and Computability Theory. The University of Western Ontario Series in Philosophy

of Science, vol 9. / Eds. R. E. Butts, J. Hintikka. — Dordrecht : Springer, 1977. — P. 149–169.

[45] *Floyd, R. W.* The paradigms of programming // Communications of the ACM. — 1979. — Vol. 22, №8. — P. 455–460.

[46] *Greibach, S. A.* New Normal-Form Theorem for Context-Free Phrase Structure Grammars // Journal of the ACM. —1965. — Vol. 12, №1. — P. 42–52.

[47] *Henner, C. R.* A Simple Set Theory for Computer Science. — Toronto, 1979. — 12 p. — (Tech. Rep. / №102.)

[48] LISP 1.5 Programming Manual / J. McCarthy, P. W. Abrahams, D. J. Edwards [et al.] — Cambridge : The MIT Press, 1963. — 106 p.

[49] *Schwartz, J. T.*. Set Theory as a Language for Program Specification and Programming. — New York, N. Y.: Courant Institute of Mathematical Sciences, 1970. — 97 p.[10]

[50] *Scott, D.* Advice on Modal Logic // Philosophical Problems in Logic, vol 29. / Ed. K. Lambert. — Dordrecht : Springer, 1970. — P. 143–173. (Русский перевод: *Скотт, Д. С.* Советы по модальной логике // Семантика модальных и интенсиональных логик / Под ред. д.ф.н. В. А. Смирнова. — М. : Прогресс, 1981. — С. 280–317.)

[51] *Scott, D. S.* Towards a mathematical semantics for computer languages / D. S. Scott, C. Strachey // Proceedings of the Symposium on Computers and Automata / Ed. J. Fox. — Brooklyn, N.Y. : Polytechnic Press, 1971. — Vol. 21. — P. 19–46.

[52] *Scott, D. S.* Logic and programming languages // Communications of the ACM. — 1977. — Vol. 20, №9. — P. 634–641. (Русский перевод: *Скотт, Д. С.* Логика и языки программирования. — Лекции лауреатов премии Тьюринга / Под ред. Р. Эшенхерста. — М. : Мир, 1993. — С. 65–83.)

[53] *Scott, D. S.* Relating theories of the lambda calculus // To H. B. Curry: Essays on combinatory logic, lambda calculus and

[10] *Данфорд, Н.* Линейные операторы : в 3 томах. / Н. Данфорд, Дж. Шварц.
Т. 1 : Общая теория. — М. : ИЛ, 1962. — 896 с.
Т. 2 : Спектральная теория. Самосопряженные операторы в гильбертовом пространстве. — М. : Мир, 1966. — 1062 с.
Т. 3 : Спектральные операторы. — М. : Мир, 1974. — 662 с.

formalism / Eds. J. P. Hindley, J. R. Seldin. — N. Y. & L.: Academic Press, 1980. — P. 403–450.

[54] *Suppes, P.* Axiomatic Set Theory. — New York, N. Y. : Dover Publications, Inc., 1972. — 267 p.

[55] *Wulf, W. A.* BLISS: A Language for Systems Programming / W. A. Wulf, D. B. Russel, A. N. Habermann // Communications of the ACM. — 1971. — Vol. 14, № 12. — P. 780–790.

### Библиографическая ссылка

*Городняя, Л. В.* От дискретной математики к семантике языков программирования // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 141–155. https://doi.org/10.26456/mfcsics-21-22

### Сведения об авторах

Лидия Васильевна Городняя
Институт систем информатики им. А. П. Ершова СО РАН. Старший научный сотрудник

*Россия, 630090, Новосибирск, пр. Академика Лаврентьева, 6*
*E-mail: lidvas@gmail.com*

УДК 510.652
AMS MSC2020: 03F30

# Интерпретации в арифметиках Бюхи[1]

**Запрягаев А. А.**

НИУ «Высшая школа экономики»

Аннотация. Арифметиками Бюхи называются расширения арифметики Пресбургера дополнительным предикатом, зависящим от натурального параметра $p$, служащие для формализации принятия множеств натуральных чисел, представленных в $p$-ичной системе счисления, конечными автоматами. В настоящей работе рассматриваются многомерные интерпретации арифметик Бюхи в себе и друг в друге для различных значений $p$. Поднимается вопрос об отсутствии иных интерпретаций арифметик Бюхи в собственных стандартных моделях, кроме определимо изоморфных тождественной. Из утвердительного ответа на указанный вопрос следует выполнение для арифметик Бюхи гипотезы Виссера, являющейся аналогом рефлексивности для слабых арифметических теорий. Устанавливается невозможность интерпретации в арифметиках Бюхи плотного порядка, откуда следует изоморфностью всякой интерпретации такого вида тождественной (не обязательно изоморфная). Также описываются интерпретации между арифметиками Бюхи для различных значений параметра $p$.

Ключевые слова: формальные арифметики, арифметики Бюхи, конечный автоматы, интерпретации, линейные порядки.

## Введение

Арифметики Бюхи (с параметром $p \in \mathbb{N}, p \geqslant 2$) называется расширение арифметики Пресбургера $\mathbf{Th}(\mathbb{N}, =, +)$ дополнительным предикатным символом $V_p(x, y)$ с семантикой «$y$ — максимальная степень $p$, на которую делится $x$» [1]. Их предложил исследовать (в несколько ином виде) Р. Бюхи в 1960 г. в качестве формализации

[1]Исследование осуществлено в рамках Программы фундаментальных исследований НИУ ВШЭ.

принятия множеств натуральных чисел конечными автоматами. В самом деле, имеет место следующее фундаментальное соответствие: Теорема 1 (V. Bruyère, 1985). *Множество $A \subseteq \mathbb{N}$, представленное в $p$-ичной системе счисления, принимается конечным автоматом тогда и только тогда, когда оно выразимо в арифметике Бюхи* $\mathbf{Th}(\mathbb{N}, =, +, V_p)$.

Мы исследуем интерпретации арифметик Бюхи в себе, а также друг в друге для различных величин параметра $p$.

Для богатых теорий, в которых возможно кодирование синтаксиса, интерес представляет понятие рефлексивности, которое вводится как способность теории доказывать непротиворечивость всех своих конечно аксиоматизируемых подтеорий. А. Виссер предложил рассматривать отсутствие интерпретаций теории в собственных конечно аксиоматизируемых подтеориях,которое следует из рефлексивности, в качестве аналога рефлексивности для слабых теорий, в том числе фрагментов арифметики Пеано. Это дает мотивировку к изучению интерпретаций арифметик Бюхи в себя. Отсутствие иных интерпретаций теории саму в себя, кроме доказуемо изоморфных тождественной, влечет предложенное свойство.

Для арифметики Пресбургера, на основе которой арифметики Бюхи определяются, доказуемый изоморфизм всех интерпретаций тождественной был ранее установлен автором и Ф. Пахомовым в работе [2].

## 1. Интерпретации

Определение 1. *Назовем $m$-мерной (непараметрической) интерпретацией $\iota$ некоторой сигнатуры первого порядка $\mathcal{K}_1$ в модели $\mathfrak{A}$ сигнатуры (возможно, другой) $\mathcal{K}_2$ совокупность следующих первопорядковых формул в сигнатуре $\mathcal{K}_2$:*

1) *$D_\iota(\overline{y})$, задающая $\mathbf{D}_\iota \subseteq \mathfrak{A}^m$ (область определения внутренней модели);*
2) *$P_\iota(\overline{x}_1, \ldots, \overline{x}_n)$ в соответствие каждому предикатному символу $P(x_1, \ldots, x_n)$ арности $n$ в сигнатуре $\mathcal{K}_1$, включая равенство;*
3) *$f_\iota(\overline{x}_1, \ldots, \overline{x}_n, \overline{y})$, в соответствие каждому функциональному символу $f(x_1, \ldots, x_n)$ арности $n$ в сигнатуре $\mathcal{K}_1$.*

*Все векторы $\overline{x}$ здесь понимаются длины $m$, а $f_\iota$ должны задавать графики функций (после факторизации по интерпретации равенства).*

Это определение легко обобщается на естественный перевод уже всех термов и формул в $\mathcal{K}_1$ на язык $\mathcal{K}_2$:

Определение 2. *Положим:*

1) $(y = f(x_1, \ldots, x_n))^\iota = f_\iota(\overline{x}_1, \ldots, \overline{x}_n, \overline{y})$;
2) $(P(x_1, \ldots, x_n))^\iota = P_\iota(\overline{x}_1, \ldots, \overline{x}_n)$;
3) *логические связки коммутируют;*
4) $(\forall x\, A(x))^\iota = \forall \overline{x}\, (D_\iota(\overline{x}) \to A^\iota)$;
5) $(\exists x\, A(x))^\iota = \exists \overline{x}\, (D_\iota(\overline{x}) \wedge A^\iota)$,

Естественным образом $\iota$ и $\mathfrak{A}$ порождают модель $\mathfrak{B}$ сигнатуры $\mathcal{K}_1$ с областью определения $\mathbf{D}_\iota/\sim_\iota$, где отношение эквивалентности $\sim_\iota$ определяется как $=_\iota (\overline{x}_1, \overline{x}_2)$. Эта $\mathfrak{B}$ называется внутренней моделью.

Определение 3. *Если $\mathfrak{B} \models \mathbf{T}$, то говорят, что $\iota$ — интерпретация теории $\mathbf{T}$ в $\mathfrak{A}$; аналогично, если даны две теории, $\mathbf{T}$ в $\mathcal{K}_1$ и $\mathbf{U}$ в $\mathcal{K}_2$, будем называть интерпретацией теории $\mathbf{T}$ в теории $\mathbf{U}$, если для всякой модели $\mathfrak{A}$ теории $\mathbf{U}$ все теоремы $\mathbf{T}$ истинны в соответствующей внутренней модели.*

Пусть теперь две интерпретации действуют в одну сигнатуру $\mathcal{K}_2$ (или теорию $\mathbf{U}$ в ней).

Определение 4. 1) *$m_1$-мерная интерпретация $\iota_1$ и $m_2$-мерная интерпретация $\iota_2$ (допустимо $m_1 \neq m_2$) называются изоморфными, если имеется изоморфизм $f$ между соответствующими внутренними моделями;*

2) *Если $f$ может при этом быть выражена $(m_1+m_2)$-местной формулой $F$ в $\mathcal{K}_2$, назовем этот изоморфизм определимым.*

Определение 5. 1) *Говорят, что интерпретация $\iota$ неотносительна, если формула $D_\iota(\overline{y})$, задающая область определения внутренней модели, тождественно истинна (областью определения является все $\mathfrak{B}^m$);*

2) *Говорят, что интерпретация $\iota$ имеет абсолютное равенство, если равенство в $\mathcal{K}_1$ интерпретируется как совпадение векторов в $\mathfrak{A}$.*

## 2. Основные результаты

Поскольку всякая арифметика Бюхи определяется как элементарная теория $(\mathbb{N}, =, +, V_n)$, не имеет значения, рассматриваются интерпретации в теории или в ее стандартной модели. Более того, без ограничения общности можно утверждать, что интерпретации всегда неотносительны и имеют абсолютное равенство:

Лемма 2. *Для любой интерпретации арифметики Бюхи в $\mathbb{N}$ как модель некоторой арифметики Бюхи существует определимо изоморфная ей неотносительная интерпретация с абсолютным равенством.*

Тем самым, всегда можно рассматривать эту новую интерпретацию вместо исходной.

Для арифметик Бюхи также имеет место известное описание порядкового типа нестандартных моделей арифметики Пеано:

Лемма 3. *Всякая нестандартная модель арифметики Бюхи, рассматриваемая как модель $(\mathbb{N}, <)$ ($<$ определимо в $\mathbb{N}, +$) имеет порядковый тип $\mathbb{N} + \mathbb{Z} \cdot \mathcal{A}$, где $\mathcal{A}$ — некоторый плотный линейный порядок без первого и последнего элементов.*

*В частности, у всякой нестандартной модели всякой арифметики Бюхи порядковый тип п л о т е н (содержит в качестве подпорядка $\mathbb{Q}$).*

Мы доказываем:

Теорема 4. *Всякий линейный порядок, определимый в некоторой арифметике Бюхи, р а з р е ж е н, то есть не содержит подпорядка, изоморфного $\mathbb{Q}$.*

Отсюда мы получаем, что при интерпретации арифметики Бюхи в $\mathbb{N}$, рассматриваемом как модель некоторой арифметики Бюхи, внутренняя модель может быть только стандартной. Тем самым:

Теорема 5. *Если интерпретация из некоторой арифметики Бюхи $\mathbf{Th}(\mathbb{N}, =, +, V_n)$ в себя существует, то она изоморфна тождественной.*

Мы выдвигаем гипотезу, что для $m = 1$ (одномерных интерпретаций) этот изоморфизм всегда определим в арифметике Бюхи $\mathbf{Th}(\mathbb{N}, =, +, V_k)$.

### 3. Интерпретации между арифметиками Бюхи для различных значений параметра

Определение 6. *Назовем числа $n, k \in \mathbb{N}$, $n, k \geqslant 2$ согласованными, если существуют $p, q \in \mathbb{N}$, $p, q > 0$ такие, что $n^p = k^q$.*

Это определение задает отношение эквивалентности на натуральных числах. Имеет место следующее:

Теорема 6. *Если числа $n, k$ согласованы, то интерпретаций из $\mathbf{Th}(\mathbb{N}, =, +, V_n)$ в $\mathbf{Th}(\mathbb{N}, =, +, V_k)$ не существует.*

*Если же $n, k$ не согласованы, то существует интерпретация из $\mathbf{Th}(\mathbb{N}, =, +, V_n)$ в $\mathbf{Th}(\mathbb{N}, =, +, V_k)$ (в силу Теоремы 5, ее внутренняя модель стандартна).*

Этот результат согласуется с теоремой Кобхэма – Семенова [1], согласно которой множество натуральных чисел, определимое в двух арифметиках Бюхи с несогласованными параметрами, определимо в арифметике Пресбургера.

### Заключение

Дальнейшие исследования в области интерпретаций арифметик Бюхи будут концентрироваться в первую очередь на явном построении определения изоморфизма при одномерной интерпретации арифметики Бюхи в собственной стандартной модели. Текущая гипотеза состоит в том, что указанное определение можно построить, что дает доказательство гипотезы Виссера в одномерном случае.

### Список литературы

[1] Logic and p-recognizable sets of integers / V. Bruyère, G. Hansel, C. Michaux, R. Villemaire // Bulletin of the Belgian Mathematical Society Simon Stevin. — 1994. — Vol. 1, №2. — P. 191–238.

[2] *Pakhomov, F.* Multi-dimensional interpretations of Presburger arithmetic in itself / F. Pakhomov, A. Zapryagaev // Journal of Logic and Computation. — 2020. — Vol. 30, №8. — P. 1681–1693.

### Библиографическая ссылка

*Запрягаев, А. А.* Интерпретации в арифметиках Бюхи // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 156–161.
https://doi.org/10.26456/mfcsics-21-23

### Сведения об авторах

Александр Александрович Запрягаев
НИУ «Высшая школа экономики». Стажер-исследователь Международной лаборатории логики, лингвистики и формальной философии, аспирант Факультета математики

*Россия, 119048, Москва, Усачева ул., д. 6, каб. 417*
*E-mail: azapryagaev@hse.ru*

УДК 519.872
AMS MSC2020: 60K25

# О стационарном распределении числа требований в одной системе массового обслуживания

**Кондратенко А. Е.**[*], **Соболев В. Н.**[**]

[*]Московский государственный университет им. М. В. Ломоносова
[**]Лаборатория ТВП

Аннотация. Рассматривается однолинейная система массового обслуживания с групповым поступлением требований, в которой длительности обслуживания имеют показательное распределение, число заявок в поступающей в систему группе требований ограничено, а число мест для ожидания не ограничено. Для данной системы массового обслуживания показано, что стационарные вероятности числа заявок в системе могут быть выражены через свертку аналогичных вероятностей вложенной цепи Маркова и нормированных хвостовых вероятностей числа заявок в поступающей в систему группы требований.

Ключевые слова: система массового обслуживания, групповое поступление, стационарное распределение, производящая функция вероятностей, вложенная цепь Маркова, процесс восстановления.

## Введение

В системе массового обслуживания $GI^{\nu}|M_{\mu}|1|\infty$ моменты поступления требований $t_0 = 0$, $t_0 < t_1 < t_2 < \ldots < t_n < \ldots$ образуют процесс восстановления [1] с функцией распределения $\mathsf{P}\{X_n < t\} = G(t)$, где $X_n = t_n - t_{n-1}$, $n \geqslant 1$.

В каждый момент $t_n$ поступает группа из $\nu_n$ требований, причем величины $\nu_n$ независимы, одинаково распределены и ограничены. Величины $\nu_n$ независимы от величин $X_n$.

В системе имеется один обслуживающий прибор, время обслуживания распределено по показательному закону с интенсивность

обслуживания $\mu$ и функцией распределения $F(t) = 1 - e^{-\mu t}$, а число мест для ожидания неограниченно.

Пусть $\xi(t)$ — число требований в системе в момент $t$. В рамках данной системы в [3], с. 171–175, (подробнее см. [5], с. 97–108) было найдено стационарное распределение процесса

$$\mathsf{P}(z) = \lim_{t\to\infty} Mz^{\xi(t)} = \sum_{n=0}^{\infty} p_n z^n,$$

в котором искомое распределение определялось через стационарное распределение вложенной однородной цепи Маркова

$$\xi_n = \xi(t_n - 0), \quad n = 1, 2, \ldots, \quad \xi_1 = 0$$

с производящей функцией

$$\pi(z) = \lim_{t\to\infty} Mz^{\xi_n} = \sum_{k=0}^{\infty} \pi_k z^k.$$

При этом выражение $\mathsf{P}(z)$ через $\pi(z)$ носило чисто аналитический характер. Однако оказывается, что данной взаимосвязи можно придать вероятностную интерпретацию. Данные результаты и будут представлены ниже.

## 1. Основные определения

Определение 1. *Определим $T$ как среднее время между поступлениями групп заявок в систему*

$$T = \mathsf{M}X_n = \int_0^{\infty} t\, dG(t). \tag{1}$$

Пусть $\alpha(z) = \mathsf{M}z^{\nu_n} = \alpha_1 z + \alpha_2 z^2 + \ldots + \alpha_m z^m$, $\alpha_m \neq 0$ — производящая функция вероятностей $\alpha_k = \mathsf{P}\{\nu_n = k\}$, $k = 1, 2, \ldots, m$. Тогда

$$\nu = \mathsf{M}\nu_n = \alpha'(z)\big|_{z=1} = \sum_{k=1}^{m} k\alpha_k \tag{2}$$

является средним числом заявок в поступающей группе.

Для входящей группы требований можно определить [4], т. 1, с. 271, вероятности хвоста распределения

$$A_k = \mathsf{P}\{\nu_n \geqslant k\} = \mathsf{P}\{\nu_n > k-1\} = \sum_{l=k}^{m} \alpha_l, \quad k = 1, \ldots, m.$$

В силу равенства

$$\sum_{k=1}^{m} A_k = \sum_{k=1}^{m} (\alpha_k + \ldots + \alpha_m) = (\alpha_1 + 2\alpha_2 + \ldots + m\alpha_m) = \nu$$

скалярные величины

$$q_k = \frac{A_k}{\nu}, \quad k = 1, \ldots, m,$$

представляют собой распределение вероятностей с производящей функцией

$$A(z) = \frac{1}{\nu} \sum_{k=1}^{m} A_k z^k = \sum_{k=1}^{m} q_k z^k. \tag{3}$$

Пусть случайная величина $Y_n$ обозначает время обслуживания $n$-й заявки. Величины $Y_n$ независимы друг от друга и от величин $X_n$, а также имеют одинаковое распределение.

В этом случае среднее время обслуживания $n$-й заявки $\tau$ конечно и равно

$$\tau = \mathsf{M}Y_n = \int_0^\infty x dF(x) = \int_0^\infty x\mu e^{-\mu x}\, dx.$$

Пусть $\eta_n$ — это число точек пуассоновского потока, с параметром $\mu$, приходящих на интервале $(t_n, t_{n+1})$. Поскольку в системе имеется один обслуживающий прибор, а время обслуживания распределено по показательному закону с параметром $\mu$, то величину $\eta_n$ можно интерпретировать как количество заявок обслуженных за время $X_{n+1} = t_{n+1} - t_n$. Среднее число обслуженных на интервале $(t_n, t_{n+1})$ требований равно

$$\mathsf{M}\eta_n = \mu T = \frac{1}{\rho_0},$$

где $\rho_0 = \lambda/\mu$. Тогда нагрузка рассматриваемой системы массового обслуживания может быть найдена по формуле

$$\rho = \nu\rho_0. \tag{4}$$

## 2. Основные результаты

Для рассматриваемого стационарного распределения справедливы (см. также [2]) следующие представления.

Теорема 1. *Если выполнено условие $\rho < 1$, то стационарное распределение процесса $\xi(t)$ существует и задается производящей функцией*

$$P(z) = \rho\pi(z)A(z) + 1 - \rho, \tag{5}$$

*где $A(z)$ определяется равенством (3), а $\rho$ — это нагрузка данной системы массового обслуживания.*

Формула (5) показывает, что искомое распределение есть смесь вырожденного распределения и распределения представляющего собой сумму распределений: стационарного распределения вложенной цепи Маркова с распределением «хвоста» входящей группы требований.

Следствие 2. *Если для системы $M^{\nu}_{\lambda}|M_{\mu}|1|\infty$ выполнено условие $\rho < 1$, то стационарное распределение процесса $\xi(t)$ существует и совпадает со стационарным распределением последовательности $\xi(t_n - 0)$, а для их производящих функцией справедливо представление*

$$P(z) = \pi(z) = \frac{1-\rho}{1-\rho A(z)}. \tag{6}$$

Формально можно определить вероятности $\pi_j$ для отрицательных индексов $j = -1, -2, \ldots, -m$. В силу того, что $\pi_j$ — это вероятность того, что в системе в момент времени $(t_n - 0)$ находится $j$ требований, то $\pi_{-1} = \ldots = \pi_{-n} = 0$.

Замечание 1. *Определение $\pi_j$ для отрицательных индексов позволяет представить стационарные вероятности $p_n$ как свертку*

$$p_n = \rho\big(A_1\pi_{n-1} + A_2\pi_{n-2} + \ldots + A_m\pi_{n-m}\big)$$

*для любого натурального $n = 1, 2, \ldots$*

**Заключение**

Введение в системе массового обслуживания $GI^{\nu}|M_{\mu}|1|\infty$ для распределения числа требований во входящей группе вероятностей хвоста данного распределения позволяет представить вероятности $p_n$ как свертку двух распределений, одно из которых — распределение вложенной цепи Маркова, а другое — описанное выше хвостовое распределение.

**Список литературы**

[1] *Гнеденко, Б. В.* Лекции по теории массового обслуживания / Б. В. Гнеденко, И. Н. Коваленко. — Киев : [б. и.], 1963. —315 с.

[2] *Соболев, В. Н.* О законе стационарной очереди для одной системы массового обслуживания с групповым поступлением требований // Управление большими системами. — 2019. — Вып. 77. — С. 6–19.

[3] *Соловьев, А. Д.* Одна система массового обслуживания с групповым поступлением требований / А. Д. Соловьев, В. Н. Соболев // Аналитические и вычислительные методы в теории вероятностей и ее приложениях (АВМТВ-2017) = Analytical and Computational Methods in Probability Theory and its Applications (ACMPT-2017) : материалы Международной научной конференции. Россия, Москва, 23–27 октября 2017, под общ. ред. А. В. Лебедева. — М. : РУДН, 2017. —С. 171–175.

[4] *Феллер, В.* Введение в теорию вероятностей и ее приложения : в 2 томах. — М. : Мир, 1964. — 2 т.

[5] *Soloviev A.D.* One Server Queue with Bulk Arrivals / A. D. Soloviev, V. N. Sobolev // Analytical and Computational Methods in Probability Theory. ACMPT 2017 (Moscow, Russia, October 23–27). Lecture Notes in Computer Science. Vol. 10684. / Eds. V. V. Rykov, N. D. Singpurwalla, A. M. Zubkov. — Cham : Springer, 2017. — P. 97–108.

**Библиографическая ссылка**

*Кондратенко, А. Е.* О стационарном распределении числа требований в одной системе массового обслуживания / А. Е. Кондратенко, В. Н. Соболев // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 162–167. https://doi.org/10.26456/mfcsics-21-24

**Сведения об авторах**

1. Александр Евгеньевич Кондратенко
Московский государственный университет им. М. В. Ломоносова. Доцент

*Россия, 119991, г. Москва, Ленинские горы, д. 1*
*E-mail: ae_cond@mech.math.msu.su*

2. Виталий Николаевич Соболев
Лаборатория ТВП. Научный сотрудник

*Россия, 117418, г. Москва, Нахимовский проспект, д. 47*
*E-mail: sobolev_vn@mail.ru*

УДК 004.93
AMS MSC2020: 68T27

# Изоморфизм предикатных формул и его применение для выделения общих свойств сложных структурированных объектов в задачах ИИ[1]

**Косовская Т. М.**

Санкт-Петербургский государственный университет

Аннотация. Под сложным структурированным объектом понимается объект, являющийся совокупностью своих элементов, каждый из которых обладает некоторыми свойствами, и эти элементы могут находиться в заданных отношениях (не обязательно бинарных). Для таких объектов адекватным языком описания является язык исчисления предикатов. Рассматривается подход к решению ряда задач распознавания и анализа сложных структурированных объектов. Введенное автором понятие изоморфизма элементарных конъюнкций предикатных формул дает возможность решать следующие задачи: создание иерархического описания множества объектов, существенно уменьшающего вычислительную сложность распознавания объектов; построение на его основе предикатных сетей для распознавания сложных структурированных объектов; построение нечетких предикатных сетей, позволяющих определить, какая часть распознаваемого объекта в какой степени похожа на часть известного объекта; «распараллеливание» процесса распознавания объекта; создание полного описания объекта при мульти-агентном сборе информации; построение онтологии множества сложных структурированных объектов.

Ключевые слова: изоморфизм предикатных формул, предикатная сеть, мульти-агентное описание, онтология.

[1]Исследование поддержано Санкт-Петербургским государственным университетом, проект № 73555239.

## Введение

Использование формул исчисления предикатов для описания и распознавания сложных структурированных объектов было предложено еще в 60-е годы XX века, например, в [7]. Под сложным структурированным объектом понимается объект, являющийся совокупностью своих элементов, каждый из которых обладает некоторыми свойствами, и эти элементы могут находиться в заданных отношениях (не обязательно бинарных).

Ранее было доказано, что при распознавании так описанных объектов возникающие задачи являются NP-трудными [1]. Однако это согласуется с тем, что несмотря на полиномиальность аналогичных задач при описании объектов строками бинарных или канечнозначных признаков, сами длины таких строк экспоненциальны от длины записи описаний на языке исчисления предикатов.

Введенное автором понятие изоморфизма элементарных конъюнкций предикатных формул (их совпадение с точностью до имен переменных и порядка конъюнктивных членов) полиномиально эквивалентно понятию изоморфизма графов [6], задача проверки которого является «открытой» задачей, для решения которой не известен полиномиальный алгоритм и не доказана ее NP-полнота.

В настоящей работе описаны задачи, решение которых основано на выделении изоморфных подформул, позволяющих находить общие свойства объектов из заданного множества.

## 1. Основные определения

Определение 1 (Сложный структурированный объект). *Сложный структурированный объект $\omega$ — это множество элементов $\omega = \{\omega_1, \ldots, \omega_t\}$, на которых задан набор предикатов $p_1, \ldots, p_n$, задающих свойства элементов и отношения между ними.*

Пример 1. *Четырехугольник, заданный четырьмя своими вершинами $\{A, B, C, D\}$, на множестве которых определены предикаты $adj(x, y)$: «$x$ смежна с $y$», $p(x, y, z, u)$: «отрезок $[x, y]$ параллелен отрезку $[z, u]$», $eq(x, y, z, u)$: «длина отрезка $[x, y]$ равна длине отрезка $[z, u]$», $ang_{sh/b/s}(x, y, z)$: «угол $xyz$ острый/тупой/прямой».*

Определение 2 (Описание сложного структурированного объекта). *Описанием сложного структурированного объекта $\omega$ называ-*

*ется элементарная конъюнкция $S(\omega)$ литералов, истинных на этом объекте.*

Пример 2. *Описанием квадрата $S(\{A, B, C, D\})$ является формула*

$$adj(A,B)\ \&\ adj(B,C)\ \&\ adj(C,D)\ \&\ \&\ adj(D,A)\ \&\ p(A,B,C,D)\ \&\ p(B,C,A,D)\ \&\ \&\ eq(A,B,C,D)\ \&\ eq(B,C,A,D)\ \&\ ang_s(A,B,C)\ \&\ \&\ ang_s(B,C,D)\ \&\ ang_s(C,D,A)\ \&\ ang_s(D,A,B).$$

Определение 3 (Описание множества объектов). *Описание множества объектов — это дизъюнкция элементарных конъюнкций с переменными в качестве аргументов, истинная для всех объектов этого множества и только для них.*

Пусть задано множество объектов $\Omega$ и его разбиение на $K$ (возможно пересекающихся) классов $\Omega = \bigcup_{k=1}^{K} \Omega_k$. $A_k(\overline{x})$[2] — описание множества $\Omega_k$. Задача идентификации, принадлежит ли объект $\omega$ множеству $\Omega_k$, сводится к проверке

$$S(\omega) \Rightarrow \exists \overline{x}_{\neq}\ A_k(\overline{x})^{3}. \qquad (1)$$

Проверка логического следования (1) является NP-полной задачей, если количество переменных в $\overline{x}$ меньше количества элементов в $\omega$ и GI-полной, если количество переменных в $\overline{x}$ равно количеству элементов в $\omega$ [1, 6]. При этом верхние оценки сложности проверки (1) экспоненциально зависят от длины записи формулы $A_k(\overline{x})$.

Определение 4 (Изоморфизм элементарных конъюнкций). *Две элементарные конъюнкции атомарных формул исчисления предикатов $P(a_1, \ldots, a_m)$ и $Q(b_1, \ldots, b_m)$ называются изоморфными $P(a_1, \ldots, a_m) \sim Q(b_1, \ldots, b_m)$, если существуют такая элементарная конъюнкция $R(x_1, \ldots, x_m)$ и подстановки аргументов $a_{i_1}, \ldots, a_{i_m}$ и $b_{j_1}, \ldots, b_{j_m}$ формул $P(a_1, \ldots, a_m)$ и $Q(b_1, \ldots, b_m)$ соответственно вместо всех вхождений переменных $x_1, \ldots, x_m$ формулы $R(x_1, \ldots, x_m)$,*

[2] $\overline{x}$ — обозначение для списка элементов конечного множества $x$, соответствующего некоторой перестановке номеров его элементов.

[3] Обозначение $\exists \overline{x}_{\neq}$ означает, что производится проверка существования набора различных значений переменных.

*что результаты этих подстановок* $R(a_{i_1}, \ldots, a_{i_m})$ *и* $R(b_{j_1}, \ldots, b_{j_m})$ *совпадают с формулами* $P(a_1, \ldots, a_m)$ *и* $Q(b_1, \ldots, b_m)$ *соответственно с точностью до порядка литералов.*

При этом полученные подстановки $(a_{i_1} \to x_1, \ldots, a_{i_m} \to x_m)$ и $(b_{j_1} \to x_1, \ldots, b_{j_m} \to x_m)$ называются унификаторами формул $P(a_1, \ldots, a_m)$ и $Q(b_1, \ldots, b_m)$ с формулой $R(x_1, \ldots, x_m)$ соответственно.

Определение 5. *Элементарная конъюнкция называется максимальной общей (с точностью до имен переменных) подформулой двух заданных элементарных конъюнкций формулой, если она изоморфна некоторым подформулам этих элементарных конъюнкций, но после добавления в нее любого литерала она не изоморфна ни одной подформуле хоть одной из них.*

Пример максимальной формулы, изоморфной подформулам двух заданных элементарных конъюнкций.

Пусть $P(a, b, c) = p_1(a, b)\ \&\ \neg p_1(a, c)\ \&\ p_2(b, c, a)$, $Q(a, b, c) = p_1(b, a)\ \&\ \neg p_1(a, c)\ \&\ p_2(a, c, b)$. Эти элементарные конъюнкции имеют две максимальные общие подформулы $R_1(a, c) = \neg p_1(a, c)$ с одним литералом и $R_2(x, y) = p_1(x, y)\ \&\ p_2(y, c, x)$, изоморфную подформулам формул $P(a, b, c)$ и $Q(a, b, c)$, так как $P(a, b, c) = R(a, b)\ \&\ p_1(a, c)$, $Q(a, b, c) = R(b, a)\ \&\ p_1(a, c)$.

Определение 6. *Максимальная общая подформула элементарных конъюнкций, задающих описания множества объектов, называется максимальным общим свойством этого множества объектов.*

## 2. Многоуровневое описание класса

Понятие максимальной формулы, изоморфной подформулам двух заданных элементарных конъюнкций, позволяет выделить из элементарных конъюнкций, входящих в $A_k(\overline{x})$, общие (с точностью до изоморфизма) подформулы [2]. Последовательное выделение общих (с точностью до изоморфизма) подформул $P_i^l(\overline{y}_i^l)$ $(l = 1, \ldots, n_l)$ из уже выделенных подформул позволяет построить многоуровневое описание класса $\Omega_k$. При этом вводятся новые предикаты $p_i^1$ и переменные $x_i^l$, определяемые равносильностями ${p_i}^l(x_i^l) \leftrightarrow P_i^l(\overline{y}_i^l)$. В элементарных конъюнкциях, входящих в $A_k(\overline{x})$, выделенные подформулы $P_i^l(\overline{y}_i^l)$ также заменяются на новые атомарные формулы

${p_i}^l(x_i^l)$.

Процесс проверки (1) сводится к последовательной проверке

$$S^l(\omega) \Rightarrow \exists \overline{y}^l_{i\neq}\ P_i^l(\overline{y}_i^l), \tag{2}$$

где $S^l(\omega)$ получена из $S^{l-1}(\omega)$ добавлением найденных постоянных формул с предикатами ${p_i}^{l-1}$. При этом длины записи $P_i^l(\overline{y}_i^l)$ меньше, чем длины записи элементарных конъюнкций, входящих в $A_k(\overline{x})$.

## 3. Предикатные сети и «распараллеливание» процесса распознавания

Процесс распознавания объекта по построенному многоуровневому описанию класса по своей сути является обходом по ориентированному графу без циклов с одной корневой вершиной, то есть по сети. Исходную сеть можно построить по обучающей выборке (достаточно долгая процедура, экспоненциально зависящая от длин записи описаний объектов) [3]. Дообучение сети на новых объектах позволяет перестраивать сеть, изменяя количество слоев в ней и количество ячеек в слое, а также устанавливая новые связи между ячейками.

Сам процесс распознавания существенно ускорится за счет того, что, во-первых, длины записи формул, проверяемых в ячейках, существенно короче длины записи исходных, во-вторых, процесс обхода сети производиться параллельно.

Содержимое ячеек предикатной сети можно видоизменить и вместо проверки (2) находить максимальную формулу, изоморфную подформулам конъюнкции литералов из $S^l(\omega)$ и $P_i^l(\overline{y}_i^l)$. При этом вычислять доли совпадения аргументов. Подробное изложение построения нечеткой сети имеется в [8].

## 4. Мульти-агентное описание объекта

Рассматривается задача построения полного описания объекта при условии, что имеется $m$ агентов, каждый из которых владеет информацией только о части этого объекта, причем настоящие имена элементов этого объекта этим агентам неизвестны и каждый из них может называть эти элементы по-своему. Однако свойства

элементов объекта и отношения между ними агентами определены правильно [4].

Очевидно, что для объединения полученных описаний в одно необходимо, чтобы части объекта, описываемые разными агентами, пересекались. Чтобы определить, что два агента описали (в качестве части своего описания) именно общую часть исследуемого объекта, требуется выделить максимальную формулу, изоморфную подформулам их описаний.

Однако этого не достаточно, так как аргументы выделенной подформулы (например, выделен треугольник на контурном изображении, вершины которого в каждом из описаний связаны еще с некоторыми точками) в каждом из описаний могут находиться в разных отношениях с остальными аргументами (например, на одном изображении в вершинах треугольника находятся петли, а на другом — висячие отрезки). В связи с этим выделенные фрагменты описаний необходимо проверять на непротиворечивость.

## 5. Построение онтологии

С точки зрения математики онтология представляет собой ориентированный граф, в корневой вершине которого находится множество объектов, а в каждой дочерней вершине графа — подмножество объектов отцовской, обладающие некоторым максимальным общим свойством.

В [9] описан способ построения онтологии, основанный на выделении максимальных общих свойств сложных структурированных объектов. Он основан на попарном выделении максимальных общих свойств объектов в каждой вершине. Если на каком-то уровне уже построенного графа появляется вершина, максимальное общее свойства объектов в этой вершине уже присутствуют в графе, то граф перестраивается таким образом, что ранее построенная вершина отождествляется с новой.

## Заключение

В работе описаны некоторые применения понятия изоморфизма элементарных конъюнкций предикатных формул к решению

различных задач ИИ, связанных с исследованием сложных структурированных объектов.

В цитируемых источниках доказаны экспоненциальные оценки сложности для всех этих задач.

**Список литературы**

[1] *Косовская, Т. М.* Доказательства оценок числа шагов решения некоторых задач распознавания образов, имеющих логические описания // Вестник Санкт-Петербургского университета. Сер. 1. — 2007. — Вып. 4. — С. 82–90.

[2] *Косовская, Т. М.* Многоуровневые описания классов для уменьшения числа шагов решения задач распознавания образов, описываемых формулами исчисления предикатов // Вестник Санкт-Петербургского университета. Сер. 10. — 2008. — Вып. 2. — С. 64–72.

[3] *Косовская, Т. М.* Самообучающаяся сеть с ячейками, реализующими предикатные формулы // Труды СПИИРАН. — 2015. — Вып. 6 (43). — С. 94–113.

[4] *Косовская, Т. М.* Мультиагентное описание сложного объекта по достоверной информации // Компьютерные инструменты в образовании. — 2016. — №4. — С. 5–18.

[5] *Косовская, Т. М.* Выделение наибольшей общей подформулы предикатных формул для решения ряда задач искусственного интеллекта / Т. М. Косовская, Д. А. Петров // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. — 2017. — Т. 13, вып. 3. — С. 250–263.

[6] *Косовская, Т. М.* Полиномиальная эквивалентность задач изоморфизм предикатных формул и изоморфизм графов / Т. М. Косовская, Н. Н. Косовский // Вестник Санкт-Петербургского университета. Математика. Механика. Астрономия. — 2019. — Т. 6, вып. 3. — С. 430–439.

[7] *Нильсон, Н.* Искусственный интеллект. Методы поиска решений. — М. : Мир, 1973. — 270 с.

[8] *Kosovskaya, T.* Fuzzy Recognition by Logic-Predicate Network // Advances in Science, Technology and Engineering Systems Journal. — 2020. — Vol. 5, №4. — P. 686–699.

[9] *Kosovskaya, T.* Extraction of isomorphic subformulas as a tool for logic ontology construction // Journal of Physics: Conference Series. — 2021. — Vol. 1864. — ID 012086.

### Библиографическая ссылка

*Косовская, Т. М.* Изоморфизм предикатных формул и его применение для выделения общих свойств сложных структурированных объектов в задачах ИИ // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 168–175.
https://doi.org/10.26456/mfcsics-21-25

### Сведения об авторах

Татьяна Матвеевна Косовская
Санкт-Петербургский государственный университет. Профессор

*Россия, 199034, Санкт-Петербург, Университетская наб., 7/9*
*E-mail: kosovtm@gmail.com*

УДК 510.649
AMS MSC2020: 06F05

# Принцип декомпозиции и алгоритмическая неразрешимость для моноидов Клини с делениями[1]

**Кузнецов С. Л.**

Математический институт им. В. А. Стеклова РАН

Аннотация. В работе рассматривается специальный класс моноидов с делениями и итерацией Клини — а именно, удовлетворяющих принципу декомпозиции итерации. Доказана $\Sigma_1^0$-полнота его инэквациональной теории; также показано, что существуют моноиды с делениями и итерацией, не удовлетворяющие принципу декомпозиции.

Ключевые слова: итерация Клини, алгоритмическая неразрешимость, алгебраическая логика.

## Введение

Понятие решетки Клини с делениями, или решетки действий (action lattice, далее РКД), введенное в работах Пратта [7] и Козена [2], соединяет в себе структуру решетки, частично упорядоченного моноида, и операцию итерации Клини. Интерес представляют атомарные, или инэквациональные теории (то есть теории атомарных неравенств) классов РКД. Известно, что инэквациональная теория всех РКД, обозначаемая **ACT**, алгоритмически неразрешима, а точнее $\Sigma_1^0$-полна [3]; для более узкого класса *-непрерывных РКД (в которых итерация Клини определяется через предельный переход, а не как неподвижная точка) соответствующая теория $\Pi_1^0$-полна [1].

[1]Работа выполнена при финансовой поддержке Совета по грантам Президента России, проект МК-1184.2021.1.1, и Российского фонда фундаментальных исследований, проект 20-01-00435.

В этой работе нас будет интересовать теория моноидов Клини с делениями (далее для краткости МКД), то есть структур, аналогичных РКД, но в которых частичный порядок не обязан быть решеткой (и, соответственно, в сигнатуре отсутствуют операции супремума и инфимума двух элементов).

В *-непрерывном случае результат о $\Pi^0_1$-полноте для МКД был доказан в работе автора [4], поэтому нас будет интересовать теория более широкого класса МКД. Более точно, данная работа представляет неразрешимость и $\Sigma^0_1$-полноту инэквациональной теории класса МКД, удовлетворяющих некоторому естественному условию, которое мы назовем принципом декомпозиции. Этот принцип верен во всех РКД и во всех *-непрерывных МКД. Однако в работе приводится пример МКД (не *-непрерывного), для которого этот принцип нарушается.

Эта работа переносит на случай с единицей (на моноиды) ранее полученные автором результаты для полугрупп с делениями и итерацией [5]. Отметим, что теории в случае полугрупп и моноидов существенно различаются (в частности, вторая не является консервативным расширением первой).

## 1. Основные определения

Определение 1 (МКД). *Моноидом Клини с делениями называется алгебраическая структура* $(\mathcal{A}; \cdot, \mathbf{1}, \preceq, /, \setminus, ^*)$, *такая что:*

- $(\mathcal{A}; \cdot, \mathbf{1})$ — *моноид;*
- $\preceq$ — *отношение частичного порядка на* $\mathcal{A}$*;*
- *операции деления* $/$ *и* $\setminus$ *связаны с операцией умножения и отношением частичного порядка следующим образом:*

$$b \preceq a \setminus c \iff a \cdot b \preceq c \iff a \preceq c / b;$$

- *итерация Клини* $a^*$ *определяется как наименьший, в смысле частичного порядка* $\preceq$*, элемент* $b$*, для которого* $\mathbf{1} \preceq b$ *и* $a \cdot b \preceq b$.

Если $\preceq$ задает структуру решетки (то есть всегда существуют $a \vee b = \sup_{\preceq}\{a, b\}$ и $a \wedge b = \inf_{\preceq}\{a, b\}$), то $\mathcal{A}$ является РКД.

Определение 2 (Теория). *Инэквациональной теорией некоторого класса $\mathcal{K}$ МКД называется множество всех общезначимых в классе $\mathcal{K}$ утверждений вида $A \preceq B$, где $A$ и $B$ — термы в сигнатуре МКД.*

Определение 3 (Принцип декомпозиции). *МКД $\mathcal{A}$ удовлетворяет принципу декомпозиции, если $\mathbf{1} \preceq b$ и $a \cdot a^* \preceq b$ влечет $a^* \preceq b$ для любых $a, b \in \mathcal{A}$.*

Легко видеть, что принципу декомпозиции удовлетворяют все РКД (за счет тождества $a^* = \mathbf{1} \vee a \cdot a^*$), а также все *-непрерывные МКД, то есть такие, что $a^* = \sup_{\preceq}\{a^n \mid n \geqslant 0\}$ для всех $a$.

## 2. Основные результаты

Теорема 1. *Инэквациональная теория класса МКД, удовлетворяющих принципу декомпозиции, $\Sigma_1^0$-полна (в частности, неразрешима).*

Доказательство. Изложим лишь общую схему доказательства. Следуя идее Бушковского [1], мы сопоставляем бесконечной работе машины Тьюринга (далее МТ) сначала тотальную (то есть задающую все непустые слова) контекстно-свободную грамматику, а затем — формулу вида $A \cdot B^* \cdot (A \cdot B^*)^* \preceq s$, где буквы исходного алфавита закодированы как $AB^i$, а сами формулы $A$ и $B$ взяты из [4] (модифицированная конструкция Сафиуллина [8]).

Собственно бесконечной работе МТ соответствует истинность этой формулы во всех *-непрерывных МКД. Однако есть МТ «застревает» в одной конфигурации («ловушке»), то формула будет истинной во всех МКД, удовлетворяющих принципу декомпозиции. Дальнейшее рассуждение, основанное на эффективной неотделимости, такое же, как и в случае решеток [3]. □

Отметим, что принцип декомпозиции в доказательстве теоремы 1 играет ключевую роль. А именно, с помощью этого принципа осуществляется разбор случаев в зависимости от длины начального отрезка исполнения МТ — достигла ли она уже «ловушки» или нет. Таким образом «разбирается» внешняя $^*$ в $(A \cdot B)^*$.

Теорема 2. *Существуют МКД, не удовлетворяющие принципу декомпозиции.*

Доказательство. Пример такого МКД можно построить явно, модифицируя конструкцию из [5]. Множество всех подмножеств

$\mathbb{N}$, с операцией (коммутативной) покомпонентного сложения, обозначаемого $\cdot$, и порядком — отношением «быть подмножеством», расширяется добавлением элементов $\xi$ и $\eta$. При этом $\xi \prec \eta$, $\mathbb{N} \prec \eta$, $\xi \succ X$, если $X \subsetneq \mathbb{N}$, а $\xi$ и $\mathbb{N}$ несравнимы. Далее, $\mathbf{1} = \{0\}$; $\xi \cdot X = \eta$ для $X \subseteq \mathbb{N}$, $X \neq \varnothing, \mathbf{1}$; $\xi \cdot \xi = \xi \cdot \eta = \eta \cdot \eta = \eta$; $\varnothing \cdot a = \varnothing$ для всех $a$.

Построенную структуру можно достроить до МКД. В частности, имеем $\{a\}^* = \mathbb{N} \not\preceq \xi$, но $\mathbf{1} \preceq \xi$ и $\{a\} \cdot \{a\}^* \preceq \xi$, что свидетельствует о нарушении принципа декомпозиции. □

Теорему 2 можно переформулировать в терминах инэквациональных теорий:

Следствие 3. *Если задать инэквациональную теорию всех МКД как расширение исчисления Ламбека с единицей* [6] *аксиомами для итерации (с правилом сечения), то в этом исчислении не будет выводимым правило декомпозиции* $\frac{\mathbf{1} \preceq B \quad A \cdot A^* \preceq B}{A^* \preceq B}$.

## Заключение

В работе рассмотрен специальный класс моноидов Клини с делениями — а именно, удовлетворяющих принципу декомпозиции. Доказана $\Sigma^0_1$-полнота его инэквациональной теории (теорема 1), а также показана его нетривиальность (теорема 2).

Вопрос об истинности теоремы 1 а также более тонкий вопрос о допустимости правила декомпозиции (в сравнении с выводимостью, следствие 3) в инэквациональной теории всех МКД остаются открытыми.

## Список литературы

[1] *Buszkowski, W.* Infinitary action logic: complexity, models and grammars / W. Buszkowski, E. Palka // Studia Logica. — 2008. — Vol. 89, № 1. — P. 1–18.

[2] *Kozen, D.* On action algebras // Logic and Information Flow / Ed. by J. van Eijck and A. Visser. — Cambridge, MA : MIT Press, 1994. — P. 78–88.

[3] *Kuznetsov, S.* Action logic is undecidable // ACM Transactions on Computational Logic. — 2021. — Vol. 2, № 10. — P. 1–26.

[4] *Kuznetsov, S.* Complexity of the infinitary Lambek calculus with Kleene star // Review of Symbolic Logic. — 2020. — P. 1–27.

[5] *Kuznetsov, S.* The 'long rule' in the Lambek calculus with iteration: undecidability without meets and joins // Proceedings of Advances in Modal Logic 2021 / Eds. N. Olivetti, R. Verbrugge, S. Negri and G. Sandu. — London : College Publications, 2020. — P. 425–449.

[6] *Lambek, J.* Deductive systems and categories II. Standard constructions and closed categories // Category Theory, Homology Theory, and their Applications I / Ed. by P. J. Hilton. — Berlin, Heidelberg : Springer, 1969. — P. 76–122.

[7] *Pratt, V.* Action logic and pure induction // Proceedings of JELIA 1990: Logics in AI / Ed. J. van Eijck. — Berlin, Heidelberg : Springer, 1991. — P. 97–120.

[8] *Сафиуллин, А. Н.* Выводимость допустимых правил с простыми посылками в исчислении Ламбека // Вестник Московского университета. Серия 1. Математика, механика. — 2007. — № 4. — С. 72–76.

### Библиографическая ссылка

*Кузнецов, С. Л.* Принцип декомпозиции и алгоритмическая неразрешимость для моноидов Клини с делениями // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 176–180.
https://doi.org/10.26456/mfcsics-21-26

### Сведения об авторах

Степан Львович Кузнецов
Математический институт им. В. А. Стеклова РАН. Старший научный сотрудник

*Россия, 119991, Москва, ГСП-1, ул. Губкина, 8, МИАН*
*E-mail: sk@mi-ras.ru*

УДК 510.6, 519.7
AMS MSC2020: 03B44

# Устранение операторов прошлого в троичной логике линейного времени на конечных трассах

**Куцак Н. Ю., Подымов В. В.**

Московский государственный университет им. М. В.Ломоносова

Аннотация. В работе рассматривается троичный вариант языка логики линейного времени (LTL), основанный на логике сильной неопределенности Клини, и предлагается подход к устранению операторов прошлого из этого языка с сохранением выразительных возможностей. Этот подход основан на особых преобразованиях троичных формул в двоичные и обратно с использованием известных результатов об устранении операторов прошлого в LTL.

Ключевые слова: темпоральная логика, логика линейного времени, троичная логика, операторы прошлого.

## Введение

Для описания и анализа поведения систем, изменяющих свои состояния во времени, широко используются языки темпоральных логик. Одни из наиболее известных темпоральных логик — это логики линейного времени (linear temporal logic, LTL) [11]. Формулы LTL строятся над переменными и константами, обозначающими истинность или ложность заданных свойств системы в конкретные моменты времени, булевыми операциями и темпоральными операторами $\mathbf{U}$ («до тех пор пока»), $\mathbf{X}$ («в следующий момент времени»).

В некоторых случаях удобно использовать и темпоральные операторы, предназначенные для выражения прошлого поведения системы: $\mathbf{U}^-$ («с тех пор как»), $\mathbf{X}^-$ («в предыдущий момент времени»). Использование операторов прошлого не увеличивает выразительности LTL, но делает формулировку свойств более интуитивной и сами формулы экспоненциально короче [8]. Однако, выразить операторы

прошлого через операторы будущего в LTL с помощью эквивалентных преобразований не представляется возможным. Методы устранения операторов прошлого из формулы устроены сложнее, чем применение одной эквивалентности для одного оператора прошлого: в [10] предложена теорема о разделении и технически непростой алгоритм перевода формулы LTL над операторами $\mathbf{U}$, $\mathbf{U}^-$, $\mathbf{X}$, $\mathbf{X}^-$ в формулу над $\mathbf{U}$, $\mathbf{X}$. В [9] предложен более эффективный алгоритм, согласно которому формула над $\mathbf{U}$, $\mathbf{U}^-$, $\mathbf{X}$, $\mathbf{X}^-$ последовательно переводится в автомат Бюхи, детерминированный автомат Мюллера, а затем счетчик автомата — в формулу LTL над $\mathbf{U}$, $\mathbf{X}$.

Ранее в [3] для спецификации диаграмм сигналов, получаемых на ранних этапах проектирования цифровых схем, мы предложили логику реального времени, названную логикой троичных сигналов (далее $\mathfrak{L}^s$). В рамках исследования выразительных возможностей $\mathfrak{L}^s$ в [1] было поставлено соответствие между $\mathfrak{L}^s$ и особой логикой дискретного времени (далее $\mathfrak{PL}^3$), предложенной нами как вариант LTL на конечных трассах [5], предназначенной для описания свойств троичных трасс с третьим истинностным значением в смысле неопределенности логики Клини [6].

В данной работе дается полное определение языка $\mathfrak{PL}^3$ (в отличие от тезисного описания в [1]) и предлагается подход к устранению операторов прошлого из $\mathfrak{PL}^3$ с сохранением выразительности языка.

## 1. Основные определения

Записью $\mathbb{N}_0$ обозначается множество всех натуральных чисел, включая ноль. Используются следующие виды интервалов: $[x, y] = \{z \mid z \in \mathbb{N}_0, x \leqslant z \leqslant y\}$; $[x, y) = \{z \mid z \in \mathbb{N}_0, x \leqslant z < y\}$; $(x, y] = \{z \mid z \in \mathbb{N}_0, x < z \leqslant y\}$.

Записью $\mathfrak{T}$ обозначено множество $\{0, *, 1\}$. Это множество используется в двух смыслах. В широком смысле, $\mathfrak{T}$ — множество значений, которыми оперируют троичные функции [4]. В узком смысле, $\mathfrak{T}$ — множество истинностных значений троичной логики Клини [6], согласно которой 0 трактуется как ложь, 1 — как истина, и $*$ — как неопределенное значение: либо истина, либо ложь, но неизвестно или неважно, что именно. Троичной функцией (местности $n$, где $n \in \mathbb{N}_0$) называется функция вида $\mathfrak{f} : \mathfrak{T}^n \to \mathfrak{T}$. На рисунке 1 изображены таблицы значений троичных функций,

| $x \diagdown y$ | $\neg x$ | $\sim x$ | $I_0(x)$ | $I_*(x)$ | $I_1(x)$ | $x \vee y$ | | | $x \& y$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | 0 | $*$ | 1 | 0 | $*$ | 1 |
| 0 | $*$ | 1 | 1 | 0 | 0 | 0 | $*$ | 1 | 0 | 0 | 0 |
| $*$ | 1 | $*$ | 0 | 1 | 0 | $*$ | $*$ | 1 | 0 | $*$ | $*$ |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | $*$ | 1 |

Рис. 1. Таблицы значений троичных функций

использующихся в работе: «зеркальное» отрицание Лукасевича ($\sim$), «циклическое» отрицание Поста ($\neg$), дизъюнкция ($\vee$), конъюнкция ($\&$) и индикаторы $I_1$, $I_0$, $I_*$. Система $\{\neg, \vee\}$ является троичным базисом [4], то есть набором функций, через которые выражаются любые троичные функции.

Двоичной функцией (местности $n$, где $n \in \mathbb{N}_0$) называется функция вида $f : (\mathfrak{T} \setminus \{*\})^n \to \mathfrak{T} \setminus \{*\}$. Таблицы значений двоичных функций получаются из таблицы 1 путем вычеркивания строк и столбцов, соответствующих значению $*$. Система $\{\sim, \vee\}$ является двоичным базисом [4].

## 2. Троичная логика линейного времени на конечных трассах

В работе представлен язык $\mathfrak{PL}^3$, введенный тезисно в [1] и являющийся «диалектом» известных языков логик линейного времени [5,7] и задающийся следующей формой Бэкуса–Наура (БНФ):

$$\varphi \ ::= \ 1 \mid * \mid p \mid \mathfrak{f}(\varphi_1, \ldots, \varphi_n) \mid \varphi_1 \mathbf{U} \varphi_2 \mid \varphi_1 \mathbf{U}^- \varphi_2,$$

где: $1, *$ —константы; $p \in \mathsf{AP}^3$ — атомарное высказывание; $\mathfrak{f}$ — троичная функция местности $n, n \in \mathbb{N}_0$; $\mathbf{U}$ — темпоральный оператор «до тех пор пока»; $\mathbf{U}^-$ — его аналог в прошлом «с тех пор как»; $\varphi_1, \ldots, \varphi_n$ — формулы.

Событием $\mathfrak{E}$ троичной трассы назовем отображение $\mathfrak{E} : \mathsf{AP}^3 \to \mathfrak{T}$. Конечную последовательность событий трассы, номер $i$-го элемента и длину этой последовательности назовем соответственно конечной троичной трассой (далее трасса), моментом времени $i$ и длиной трассы.

Каждым формуле $\varphi$, трассе $\pi = (\mathfrak{E}_0, \mathfrak{E}_1, \ldots, \mathfrak{E}_k)$, $k \in \mathbb{N}_0$, и моменту времени $i$, $i \in [0, k]$, сопоставим значение $\varphi[\pi, i]$ множества $\mathfrak{T}$, которое назовем значением формулы $\varphi$ на трассе $\pi$ в момент времени $i$. Определим также значение $\varphi[\pi, \mathcal{I}]$ формулы $\varphi$ на интервале натуральных чисел $\mathcal{I}$: $\varphi[\pi, \mathcal{I}] = v$ ($\varphi[\pi, \mathcal{I}] \neq v$), $v \in \mathfrak{T}$, если для любого момента времени $i$, $i \in \mathcal{I}$, верно $\varphi[\pi, i] = v$ ($\varphi[\pi, i] \neq v$).

Семантику формул $\mathfrak{PL}^3$ определим индуктивно по построению формулы следующими правилами:

$$c[\pi, i] = c, \text{ где } c \in \{1, *\}.$$

$$p[\pi, i] = \mathfrak{E}_i(p), \text{ если } p \in \mathsf{AP}^3.$$

$$\mathfrak{f}(\varphi_1, \ldots, \varphi_n)[\pi, i] = \mathfrak{f}(\varphi_1[\pi, i], \ldots, \varphi_n[\pi, i]).$$

$$(\varphi_1 \mathbf{U} \varphi_2)[\pi, i] = \begin{cases} 1, & \text{если } \exists i', i' \geqslant i : \varphi_2[\pi, i'] = \varphi_1[\pi, [i, i')] = 1; \\ *, & \text{если условие выше неверно и} \\ & \quad \exists i', i' \geqslant i : \varphi_2[\pi, i'] \neq 0 \text{ и } \varphi_1[\pi, [i, i')] \neq 0; \\ 0, & \text{иначе.} \end{cases}$$

$$(\varphi_1 \mathbf{U}^- \varphi_2)[\pi, i] = \begin{cases} 1, & \text{если } \exists i', i' \leqslant i : \varphi_2[\pi, i'] = \varphi_1[\pi, (i', i]] = 1; \\ *, & \text{если условие выше неверно и} \\ & \quad \exists i', i' \leqslant i : \varphi_2[\pi, i'] \neq 0 \text{ и } \varphi_1[\pi, (i', i]] \neq 0; \\ 0, & \text{иначе.} \end{cases}$$

Формулы $\varphi_1$ и $\varphi_2$ инициально эквивалентны, если на каждой возможной трассе $\pi$ верно $\varphi_1[\pi, 0] = \varphi_2[\pi, 0]$, обозначается $\varphi_1 =_i \varphi_2$.

Формулу $\mathfrak{L}^3$ определим как формулу $\mathfrak{PL}^3$ без оператора $\mathbf{U}^-$. Формулы $\mathfrak{PL}^2$ и $\mathfrak{L}^2$ определим соответственно как формулы $\mathfrak{PL}^3$ и $\mathfrak{L}^3$, в которых из синтаксиса и семантики удалены все части, относящиеся к значению $*$, и троичные функции заменены двоичными. Отметим, что язык $\mathfrak{L}^2$ совпадает с языком $\mathrm{LTL}_f$ [5], из которого удален оператор $\mathbf{X}$.

## 3. Основные результаты

В данном разделе предлагается проиллюстрированный на рис. 2 способ преобразования произвольной формулы $\mathfrak{PL}^3$ в инициально эквивалентную формулу $\mathfrak{L}^3$. Без потери общности рассуждений далее полагаем, что из всех троичных и двоичных функций в языках

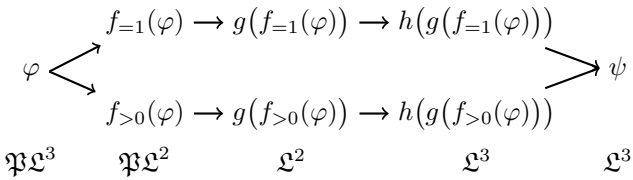



Рис. 2. Схема устранения оператора $\mathbf{U}^-$ из формулы $\mathfrak{PL}^3$

$\mathfrak{PL}^3, \mathfrak{L}^3, \mathfrak{PL}^2$ и $\mathfrak{L}^2$ используются только функции упомянутых выше троичного и двоичного базисов, а другие функции расцениваются как соответствующие сокращения.

Рассмотрим множество $\mathsf{AP}^2 = \{p_{=1} \mid p \in \mathsf{AP}^3\} \cup \{p_{>0} \mid p \in \mathsf{AP}^3\}$. Для троичной трассы $\pi^3 = (\mathfrak{E}_0^3, \mathfrak{E}_1^3, \ldots, \mathfrak{E}_k^3)$ определим взаимно соответствующую двоичную трассу $\pi^2 = (\mathfrak{E}_0^2, \mathfrak{E}_1^2, \ldots, \mathfrak{E}_k^2)$, где $\mathfrak{E}_i^2 : \mathsf{AP}^2 \to \mathfrak{T} \setminus \{*\}, i \in [0, k]$, следующим образом:

$$\mathfrak{E}_i^2(p_{=1}) = \begin{cases} 1, \text{если } \mathfrak{E}_i^3(p) = 1; \\ 0, \text{иначе}; \end{cases} \qquad \mathfrak{E}_i^2(p_{>0}) = \begin{cases} 0, \text{если } \mathfrak{E}_i^3(p) = 0; \\ 1, \text{иначе}. \end{cases}$$

Для формулы $\varphi$ языка $\mathfrak{PL}^3$ определим две соответствующие формулы $f_{=1}(\varphi)$ и $f_{>0}(\varphi)$ языка $\mathfrak{PL}^2$:

$$\begin{aligned}
&f_{=1}(1) = 1, && f_{>0}(1) = 1,\\
&f_{=1}(*) = 0, && f_{>0}(*) = 1,\\
&f_{=1}(p) = p_{=1}, && f_{>0}(p) = p_{>0},\\
&f_{=1}(\neg\varphi) = \sim (f_{=1}(\varphi) \vee \sim f_{>0}(\varphi)), && f_{>0}(\neg\varphi) = \sim f_{=1}(\varphi),\\
&f_{=1}(\varphi_1 \vee \varphi_2) = f_{=1}(\varphi_1) \vee f_{=1}(\varphi_2), && f_{>0}(\varphi_1 \vee \varphi_2) = f_{>0}(\varphi_1) \vee f_{>0}(\varphi_2),\\
&f_{=1}(\varphi_1 \mathbf{U} \varphi_2) = f_{=1}(\varphi_1) \mathbf{U} f_{=1}(\varphi_2), && f_{>0}(\varphi_1 \mathbf{U} \varphi_2) = f_{>0}(\varphi_1) \mathbf{U} f_{>0}(\varphi_2),\\
&f_{=1}(\varphi_1 \mathbf{U}^- \varphi_2) = f_{=1}(\varphi_1) \mathbf{U}^- f_{=1}(\varphi_2), && f_{>0}(\varphi_1 \mathbf{U}^- \varphi_2) = f_{>0}(\varphi_1) \mathbf{U}^- f_{>0}(\varphi_2).
\end{aligned}$$

Лемма 1. *Для любой формулы $\varphi$ языка $\mathfrak{PL}^3$, соответствующих друг другу троичной и двоичной трасс $\pi^3$ и $\pi^2$ длины $k+1$ и момента времени $i$, $i \in [0,k]$, верно*

$$\varphi[\pi^3,i]=1 \Leftrightarrow f_{=1}(\varphi)[\pi^2,i]=1,$$
$$\varphi[\pi^3,i]\neq 0 \Leftrightarrow f_{>0}(\varphi)[\pi^2,i]=1.$$

Доказательство. Доказательство проводится индуктивно по построению формулы $\mathfrak{PL}^3$. Истинность леммы следует из определений семантических правил формул $\mathfrak{PL}^3$, $\mathfrak{PL}^2$ и функций $f_{=1}$, $f_{>0}$.

Подробно рассмотрим случай $\varphi=\neg\varphi_1$, остальные доказываются аналогично. По определению отрицания Поста верно

$$(\neg\varphi_1)[\pi^3,i]=1 \Leftrightarrow \varphi_1[\pi^3,i]=* \Leftrightarrow \varphi_1[\pi^3,i]\neq 1 \text{ и } \varphi_1[\pi^3,i]\neq 0.$$

По индуктивному предположению для $\varphi_1$ верно

$$\varphi_1[\pi^3,i]\neq 1 \Leftrightarrow f_{=1}(\varphi)[\pi^2,i]\neq 1 \Leftrightarrow f_{=1}(\varphi)[\pi^2,i]=0;$$
$$\varphi_1[\pi^3,i]\neq 0 \Leftrightarrow f_{>0}(\varphi)[\pi^2,i]=1.$$

По определению функции $f_{=1}$ верно

$$f_{=1}(\neg\varphi_1)[\pi^2,i]=(\sim(f_{=1}(\varphi_1)\vee \sim f_{>0}(\varphi_1)))[\pi^2,i],$$

а значит,

$$f_{=1}(\neg\varphi_1)[\pi^2,i]=1 \Leftrightarrow f_{=1}(\varphi_1)[\pi^2,i]=0 \text{ и } f_{>0}(\varphi_1)[\pi^2,i]=1,$$

то есть выполнена первая равносильность из условия леммы, вторая доказывается аналогично. □

Теорема 2 ([10]). *Для любой формулы $\varphi$ языка $\mathfrak{PL}^2$ существует инициально эквивалентная формула $g(\varphi)$ языка $\mathfrak{L}^2$, то есть*

$$\varphi[\pi^2,0]=1 \Leftrightarrow g(\varphi)[\pi^2,0]=1.$$

Преобразуем формулы $\mathfrak{L}^2$ в формулы $\mathfrak{L}^3$ по правилу $h$:

$$h(1)=1, \qquad h(\sim\varphi)=I_0(\varphi)\vee\neg I_1(\varphi),$$
$$h(p_{=1})=I_1(p), \qquad h(\varphi_1\vee\varphi_2)=h(\varphi_1)\vee h(\varphi_2),$$
$$h(p_{>0})=\sim I_0(p), \qquad h(\varphi_1\mathbf{U}\varphi_2)=h(\varphi_1)\mathbf{U}h(\varphi_2).$$

Лемма 3. *Для любой формулы $\varphi$ языка $\mathfrak{L}^2$, соответствующих друг другу двоичной и троичной трасс $\pi^2$ и $\pi^3$ длины $k+1$ и момента времени $i$, $i \in [0,k]$, верно*

$$\varphi[\pi^2, i] = 1 \Leftrightarrow h(\varphi)[\pi^3, i] = 1,$$
$$\varphi[\pi^2, i] = 0 \Leftrightarrow h(\varphi)[\pi^3, i] = 0.$$

Доказательство. Следует из определения соответствия троичной и двоичной трассы и эквивалентностей для троичных функций. □

Лемма 4. *Для любых формулы $\varphi$ языка $\mathfrak{PL}^3$ и трассы $\pi^3$ верно*

$$\varphi[\pi^3, 0] = 1 \Leftrightarrow (\chi_{=1} \vee *\&\chi_{>0})[\pi^3, 0] = 1,$$
$$\varphi[\pi^3, 0] \neq 0 \Leftrightarrow (\chi_{=1} \vee *\&\chi_{>0})[\pi^3, 0] \neq 0,$$

*где* $\chi_{=1} = h\big(g\big(f_{=1}(\varphi)\big)\big)$, $\chi_{>0} = h\big(g\big(f_{>0}(\varphi)\big)\big)$.

Доказательство. Пусть троичная трасса $\pi^3$ и двоичная трасса $\pi^2$ соответствуют друг другу. Из лемм 1 и 3, теоремы 2 следуют цепочки равносильностей:

$$\varphi[\pi^3, 0] = 1 \Leftrightarrow f_{=1}(\varphi)[\pi^2, 0] = 1 \Leftrightarrow g\big(f_{=1}(\varphi)\big)[\pi^2, 0] = 1 \Leftrightarrow$$
$$\Leftrightarrow \chi_{=1}[\pi^3, 0] = 1 \Leftrightarrow (\chi_{=1} \vee *\&\chi_{>0})[\pi^3, 0] = 1;$$
$$\varphi[\pi^3, 0] \neq 0 \Leftrightarrow f_{>0}(\varphi)[\pi^2, 0] = 1 \Leftrightarrow g\big(f_{>0}(\varphi)\big)[\pi^2, 0] = 1 \Leftrightarrow$$
$$\Leftrightarrow \chi_{>0}[\pi^3, 0] = 1 \Leftrightarrow (\chi_{=1} \vee *\&\chi_{>0})[\pi^3, 0] \neq 0. \quad \square$$

Из леммы 4 напрямую вытекает основной результат работы — следующая теорема.

Теорема 5. *Для любой формулы языка $\mathfrak{PL}^3$ существует инициально эквивалентная формула языка $\mathfrak{L}^3$.*

## Заключение

Мы показали, что языки троичных логик линейного времени без оператора **X** на конечных трассах с операторами прошлого ($\mathfrak{PL}^3$) и без них ($\mathfrak{L}^3$) одинаково выразительны. В дальнейшем предполагается совместить этот результат с полученными ранее в [1, 2], чтобы показать, что подавляющая часть темпоральных операторов логики троичных сигналов [3] может быть удалена из языка с сохранением выразительных возможностей.

**Список литературы**

[1] *Куцак, Н. Ю.* Дискретизация сигнальной логики // ЛОМОНОСОВ-2021. Сборник тезисов XXVIII Международной научной конференции студентов, аспирантов и молодых ученых / Отв. ред. И. А. Алешковский [et al.] — М. : МАКС Пресс, 2021. — С. 102–104.

[2] *Куцак, Н. Ю.* О выразимости операций логики троичных цифровых сигналов / Н. Ю. Куцак, В. В. Подымов // Научная конференция «Тихоновские чтения 2020»: Тезисы докладов. — 2020. — URL: https://istina.msu.ru/conferences/presentations/330784820/. — Загл. с титул. экрана.

[3] *Куцак, Н. Ю.* Формальная верификация диаграмм троичных цифровых сигналов / Н. Ю. Куцак, В. В. Подымов // Моделирование и анализ информационных систем. — 2019. — Т. 26, №3. — С. 332–350.

[4] *Яблонский, С. В.* Введение в дискретную математику. — М. : Наука, 1986. — 384 с.

[5] *De Giacomo, G.* Linear temporal logic and Linear Dynamic Logic on finite traces / G. De Giacomo, M. Vardi // International Joint Conference on Artificial Intelligence (IJCAI). — 2013. — P. 854–860.

[6] *Kleene, S. C.* On notation for ordinal numbers // The Journal of Symbolic Logic. — 1938. — Vol. 3, №4. — P. 150–155.

[7] *Konikowska, B.* A Three-Valued Linear Temporal Logic for Reasoning about Concurrency. — Warsaw, Poland, 1998. — 9 p. — (Tech. Rep. / ICS PAS; 01-237.)

[8] *Kupferman, O.* Once and for all / O. Kupferman, A. Pnueli, M. Vardi // Journal of Computer and System Sciences. — 2012. — Vol. 78, №3. — P. 981–996.

[9] *Markey, N.* Temporal Logic with Past is Exponentially More Succinct // Bulletin of the EATCS. — 2003. — Vol. 79. — P. 122–128.

[10] On the temporal analysis of fairness / D. Gabbay, A. Pnueli, S. Shelah, J. Stavi // In Conference Record of the 7th Annual ACM Symposium on Principles of Programming Languages (POPL'80). — New York, N. Y. : ACM Press, 1980. — P. 163–173.

[11] *Pnueli, A.* The temporal logic of programs // In Proceedings of the 18th Annual IEEE Symposium on Foundations of Computer Science (FOCS'77). — Los Alamitos, CA : IEEE Computer Society Press, 1977. — P. 46–57.

### Библиографическая ссылка

*Куцак, Н. Ю.* Устранение операторов прошлого в троичной логике линейного времени на конечных трассах / Н. Ю. Куцак, В. В. Подымов // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 181–189.
https://doi.org/10.26456/mfcsics-21-27

### Сведения об авторах

1. Нина Юрьевна Куцак
Московский государственный университет им. М. В.Ломоносова. Аспирант

*Россия, 119991, Москва, Ленинские горы, д.1, стр. 52, 2-й учебный корпус, факультет ВМК, кафедра МК*
*E-mail: nina_svetik@mail.ru*

2. Владислав Васильевич Подымов
Московский государственный университет им. М. В.Ломоносова. Доцент

*Россия, 119991, Москва, Ленинские горы, д.1, стр. 52, 2-й учебный корпус, факультет ВМК, кафедра МК*
*E-mail: valdus@yandex.ru*

УДК 519.21
AMS MSC2020: 60J80, 81Q15, 65C05

# Моделирование процессов с генерацией и транспортом частиц в случайной среде[1]

**Куценко В. А., Яровая Е. Б.**

Московский государственный университет им. М. В. Ломоносова

Аннотация. Рассматриваются различные модели ветвящегося случайного блуждания с непрерывным временем по многомерной решетке. В основе процесса лежит симметричное, однородное по пространству, неприводимое случайное блуждание с конечной дисперсией скачков. Интенсивности размножения и гибели частиц в точках решетки предполагаются случайными. Для ветвящихся случайных блужданий в случайных средах характерно влияние редких флуктуаций, поэтому осредненное описание, типичное при классическом подходе, не всегда адекватно. В частности, для процессов в случайных средах характерно возникновение нерегулярных структур с выраженной неоднородностью пространственного распределения. В физической литературе для подобных явлений принят термин «перемежаемость». На основе результатов моделирования удалось показать, что эффект перемежаемости может наблюдаться и быть численно оценен в случайных средах даже на конечных временных интервалах.

Ключевые слова: ветвящееся случайное блуждание, многомерная решетка, случайная ветвящаяся среда, моменты численностей частиц, моделирование.

## Введение

В этой работе основное внимание уделяется моделированию ветвящихся случайных блужданий (ВСБ) в случайной среде и, в частности, анализу явления перемежаемости моментов численностей частиц. Перемежаемость есть аномальное свойство предельного распределения поля моментов, возникающее в случайной среде. В [1, 5]

[1]Работа выполнена при финансовой поддержке РФФИ, проект 20-01-00487.

явление перемежаемости было изучено для стационарной случайной среды на примере задачи Коши для оператора Андерсона со случайным потенциалом:

$$\partial u/\partial t = \varkappa \Delta u + V(x,\omega)u, \qquad u(t,x)|_{t=0} \equiv 1. \tag{1}$$

Здесь оператор $\Delta$, задающий «диффузию» частиц, действует как разностный лапласиан на $\mathbf{Z}^d$: $(\Delta u)(x) = \sum_{x':|x-x'|=1}(u(x') - u(x))$, где $\varkappa > 0$ — коэффициент диффузии, а потенциал $V(x,\omega)$, отвечает наличию ветвления и представляет собой совокупность гауссовских независимых одинаково распределенных случайных величин. В [4] с помощью применения формулы Фейнмана – Каца доказаны существование и единственность решения уравнения (1) для некоторого класса неотрицательных функций на $\mathbf{Z}^d$ и дано строгое математическое определение перемежаемости. В работе [2] для ВСБ в однородной случайной среде получены предельные теоремы для моментов численности частиц и установлена перемежаемость поля моментов. Наконец, в [7] результаты обобщены для произвольных симметричных случайных блужданий с конечной дисперсией скачков и конечным числом центров генерации частиц на $\mathbf{Z}^d$.

Большинство результатов, полученных для ВСБ в случайных средах, являются асимптотическими. В то же время изучение ВСБ на конечных интервалах времени представляется сложной задачей, которая, насколько нам известно, ранее не исследовалась. В связи с этим, основная цель численного моделирования — продемонстрировать возможность получения результатов, предсказанных теоретически, за конечное время. Аналогичная задача рассматривалась в [3] для ВСБ в неслучайных средах. Первый шаг в подобном исследовании ВСБ в случайных средах был предпринят, по-видимому, в [6], где удалось показать наличие эффекта перемежаемости в случайных средах на конечных временных интервалах. В данной работе исследуются эффекты перемежаемости в зависимости от длины временного интервала и вводится способ его численной оценки.

## 1. Основные определения

Пусть в каждом узле решетки $x \in \mathbb{Z}^d$, $d \in \mathbb{N}$, определен процесс рождения и гибели частиц. Соответствующие интенсивности задаются неотрицательными случайными величинами $\xi^+(x) = \xi^+(x,\omega)$ и $\xi^-(x) = \xi^-(x,\omega)$, определенными на вероятностном пространстве $(\Omega, \mathcal{F}, \mathbb{P})$. Математическое ожидание относительно меры $\mathbb{P}$ будет обозначаться $\langle\cdot\rangle$. Среда (то есть набор характеристик ветвления в точках решетки, называемых *источниками ветвления*) представляет собой совокупность пар случайных величин $(\xi^+(x), \xi^-(x))$. Будем предполагать, что пары $(\xi^+(x), \xi^-(x))$ независимы и одинаково распределены. При фиксированном $\omega \in \Omega$ (то есть при фиксированной реализации среды) механизм ветвления задается марковским процессом ветвления. Такая ветвящаяся среда называется *однородной*. Если ветвящийся процесс происходит в лишь конечном числе точек $x \in \mathbb{Z}^d$ и задается случайными величинами $\xi(x_i) := \xi(x_i,\omega) = (\xi^-(x_i,\omega), \xi^+(x_i,\omega))$, $i = 1, 2, \ldots, N$, то такая ветвящаяся среда называется *неоднородной*.

Транспорт частиц в моделях ВСБ описывается симметричным, однородным по пространству, неприводимым случайным блужданием, которое задается инфинитезимальной матрицей перехода $A = (a(x,y))_{x,y\in\mathbb{Z}^d}$, см. [7]. Однако в этой работе мы ограничимся рассмотрением простого симметричного случайного блуждания, в котором частица ждет время $Exp(\varkappa)$, а затем равновероятно перемещается в одну из соседних точек решетки. В [2] показано, что ВСБ с таким блужданием обладает всеми интересующими нас свойствами, в частности, перемежаемостью. Предположим, что в момент времени $t = 0$ на решетке находится ровно одна частица в точке $x \in \mathbb{Z}^d$, которая являлась источником, то за время $[0, h)$, при $h \to 0$ частица может: прыгнуть в соседнюю точку $y$ с вероятностью $\frac{\varkappa}{2d}h + o(h)$, произвести одного потомка с вероятностью $\xi^+(x)h + o(h)$, умереть с вероятностью $\xi^-(x)h + o(h)$, или, наконец, выжить без изменений с вероятностью $1 - \varkappa h - (\xi^+(x) + \xi^-(x))h + o(h)$. Если же точка $x \in \mathbb{Z}^d$ не была источником ветвления, то за время $[0, h)$, при $h \to 0$, частица может прыгнуть в соседнюю точку $y$ с вероятностью $p(h,x,y) = \frac{\varkappa}{2d}h + o(h)$ или остаться в точке $x$ без изменений с вероятностью $1 - \varkappa h - (\xi^+(x) + \xi^-(x))h + o(h)$. Эволюция частиц происходит независимо друга от

друга и от всей предыстории. Состояние системы частиц на $\mathbb{Z}^d$ описывается числом частиц $\mu_{t,\omega}(y)$ в момент времени $t$ в точке $y \in \mathbb{Z}^d$, а также общим числом частиц $\mu_{t,\omega} := \sum_{y\in\mathbb{Z}^d} \mu_{t,\omega}(y)$ на $\mathbb{Z}^d$ при начальных условиях $\mu_{0,\omega}(y) = \delta_y(x)$ и $\mu_{0,\omega} = 1$, соответственно. Для описания поведения $\mu_{t,\omega}(y)$ и $\mu_{t,\omega}(y)$ прибегают к вычислению их моментов. Первые «замороженные» моменты (quenched moments) (см. [2]) являются случайными и определяются как $m_1(t,x,y) := m_1(t,x,y,\omega) = \mathbb{E}_x\mu_{t,\omega}(y)$; $m_1(t,x) := m_1(t,x,\omega) = \mathbb{E}_x\mu_{t,\omega}$. Здесь $\omega$ относится к фиксированной («замороженной») реализации случайной среды, а $x$ есть положение начальной частицы при $t = 0$. Первые «отожженые» моменты (annealed moments) определяются как $\langle m_1(t,x,y)\rangle$ и $\langle m_1(t,x)\rangle$, соответственно.

В неслучайной среде интенсивности ветвления предполагаются равными константе см., напр., [3] и библиографию в ней. В остальном, описание процесса такое же, как и в случайной среде. Однако в неслучайной среде нет понятия «замороженного момента». Моменты численностей частиц $m_1(t,x,y) = \mathbb{E}_x\mu_t(y)$ и $m_1(t,x) = \mathbb{E}_x\mu_t$, как в классической теории, неслучайны.

Как мы уже упоминали, перемежаемость есть аномальное свойство предельного распределения поля моментов. В общем случае определение этого понятия достаточно сложно, см. [2,4]. Однако в рамках данной работы, перемежаемость можно определить как быстрый рост моментов: $\langle m^2\rangle \gg \langle m\rangle^2$, $\langle m^4\rangle \gg \langle m^2\rangle^2$ и т. д. Ключевой результат, показанный в [2,7] гласит, что при некоторых достаточно общих ограничениях на интенсивности для отожженных моментов ВСБ в однородной и неоднородной среде поле замороженных моментов перемежаемо. В [6] с помощью численного моделирования продемонстрировано, что основной вклад в каждый отожженый момент вносят высокие и редкие «пики» случайного поля замороженных моментов. В данной работе исследуется возникновение перемежаемости в зависимости от длины временного интервала и вводится способ ее оценки.

## 2. Основные результаты

Для численного моделирования использовалась среда R v 3.6.3 и компьютерный кластер с 48 процессорными ядрами. Алгоритм

использовал разложение ВСБ на композицию полиномиальных и экспоненциальных случайных величин. Подробное описание алгоритма см. в [6]. Этот алгорим позволяет генерировать значение общей численности $\mu_{t,\omega}$ на интервале времени $[0, T]$ для заранее заданных параметров модели и реализации среды $\omega$.

Пусть выполнено $M$ запусков алгоритма и получены численности $\mu_{t,\omega,1}, \ldots, \mu_{t,\omega,M}$. Тогда оценить $m_1(t,\omega) = \mathbb{E}\mu_{t,\omega}$ можно с помощью метода Монте-Карло, положив $\hat{m}_1(t,\omega) = \dfrac{1}{M}\sum\limits_{i=1}^{M} \mu_{t,\omega,i}$. Пусть значения $\hat{m}_1(t,\omega_1), \ldots, \hat{m}_1(t,\omega_{M_1})$ оценены для различных $(\omega_1, \ldots, \omega_{M_1})$, соответственно. Тогда оценка отожженных моментов $\langle m_1(t)\rangle$ может быть вычислена как $\widehat{\langle m_1(t)\rangle} = \dfrac{1}{M_1}\sum\limits_{k=1}^{M_1} \hat{m}_1(t,\omega_k)$.

В неслучайной среде значение $m_1(t)$ не является случайным, поэтому метод Монте-Карло может быть остановлен на первом шаге с моделированием $M$ оценок. Однако для удобства сравнения случайных и неслучайных сред мы использовали «псевдотожженный» момент $\widehat{[m_1(t)]}$: $\widehat{[m_1(t)]} = \dfrac{1}{M_1}\sum\limits_{k=1}^{M_1} \hat{m}_1(t)$.

При моделировании мы рассматривали простое случайное блуждание с $\varkappa = 1$ и полагали $T = 10$, $M = 1000$ и $M_1 = 250$. В ВСБ в неслучайной среде для однородного и неоднородного случаев интенсивность гибели принималась равной 1, а интенсивность деления частиц — равной 2. ВСБ в случайной среде рассматривается для тех же случаев, но интенсивности описываются вейбулловскими случайными величинами с параметрами гибели $\mathbb{E}\,(\mathrm{Weib}(2, 1.13)) \approx 1$ и деления $\mathbb{E}\,(\mathrm{Weib}(2, 2.26)) \approx 2$ для удобства сравнения с неслучайной средой. На рисунке 1 показаны оценки первого момента общего числа частиц для этих моделей в моменты времени $t = 1$ и $t = 2$, при которых уже визуально выражена перемежаемость в однородной и неоднородной случайных средах. В неслучайной среде гипотезу о том, что оценки первого момента отклоняются от реализаций нормально распределенной случайной величины с помощью теста Шапиро – Уилка отвергнуть не удалось ($p > 0.1$ в обоих случаях). При этом в случайной среде распределение замороженных моментов содержит редкие реализации большого размера.

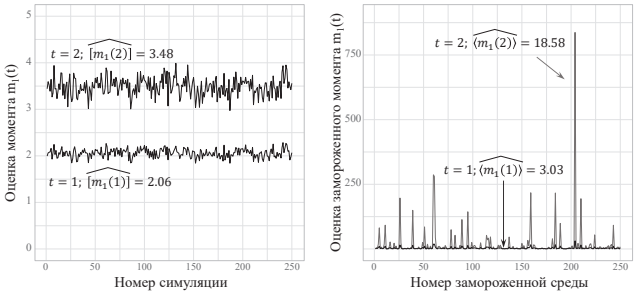



Рис. 1. Моменты для ВСБ в неоднородной среде неслучайной (слева) и случайной среде

Введем «меру» перемежаемости $R_k(t)$, несколько модифицированную по сравнению с введенной в [6]. Мера основана на интерпретации перемежаемости — как появлении «высоких пиков», влияние которых оценивается при помощи усеченного среднего. Значение первого момента численностей частиц, усеченного на уровне $k\%$, определяется как среднее, оцененное по выборке, без наименьших и наибольших $k\%$ наблюдений.

$$R_k(t) = \begin{cases} \widehat{[m_1(t)]} \Big/ \widehat{[m_1(t)]}_{k\%\ \text{усеч.}} \text{ в неслучайной среде,} \\ \widehat{\langle m_1(t)\rangle} \Big/ \widehat{\langle m_1(t)\rangle}_{k\%\ \text{усеч.}} \text{ в случайной среде.} \end{cases}$$

В неслучайных средах $R_1(t)$ и $R_5(t)$ для каждого $0 < t \leqslant 10$ равны 1 с точностью до второго знака после запятой. В то же время в случайной средах $R_1(t)$ и $R_5(t)$, как видно из рисунка 2, резко отличаются от единицы и $R_1(t) < R_5(t)$ для каждого $0 < t \leqslant 10$. Мы отдельно оценили монотонность функции $R_k(10)$ по $k$ на равномерной решетке из 100 точек на отрезке $k \in [0.21, 5]$. Оказалось, что $R_k(10)$ монотонно возрастает при увеличении $k$. При минимальном урезании на одну траекторию $k = 0.21$ и $R_{0.21}(10) = 79 > 1$. Численно показано, что величина $R_k(10) \approx 1$ при $k \in [0.21, 5]$ для неслучайной среды, а для случайной среды $R_k(10)$ резко отличается от 1 и монотонно возрастает по $k$. Тот же эффект наблюдается для однородной «критической» случайной среды, когда интенсивности

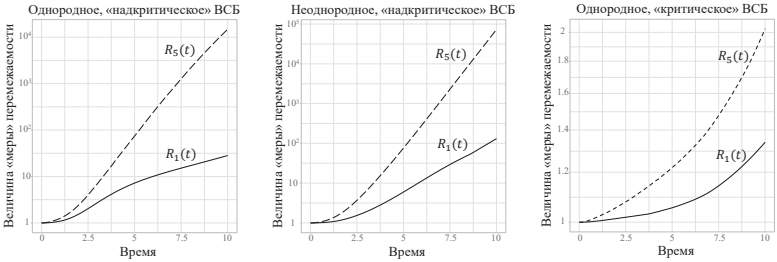



Рис. 2. «Мера» перемежаемости $R_k(t)$ для различных моделей

размножения и гибели частиц имеют одинаковое распределение с параметрами $\mathrm{Weib}(2, 1.13)$.

## Заключение

Перемежаемость может наблюдаться на конечных временных интервалах. С помощью функции $R_k(t)$ для каждого фиксированного $k$ численно оценивается величина эффекта перемежаемости, которая увеличивается с ростом $t$ в случайных средах.

## Список литературы

[1] Перемежаемость в случайной среде / Я. Б. Зельдович, С. А. Молчанов, А. А. Рузмайкин, Д. Д. Соколов // Успехи физических наук. — 1987. — Т. 152, вып. 1. — С. 3–32.

[2] Annealed Moment Lyapunov Exponents for a Branching Random Walk in a Homogeneous Random Branching Environment / S. Albeverio, L. Bogachev, S. Molchanov, E. Yarovaya // Markov Processes and Related Fields. — 2000. — №6. — P. 473–516.

[3] *Ermishkina, E.* Simulation of Branching Random Walks on a Multidimensional Lattice / E. Ermishkina, E. Yarovaya // Journal of Mathematical Sciences. — 2021. — Vol. 254, №4. — P. 469–484.

[4] *Gärtner J.* Parabolic problems for the Anderson model / J.Gärtner, S. A. Molchanov // Communications in Mathematical Physics. — 1990. — №132. — P. 613–655.

[5] Intermittency, diffusion and generation in a nonstationary random medium / I. B. Zel'dovich, S. A. Molchanov, A. A. Ruzmaikin, D. D. Sokolov // Cambridge : Cambridge Scientific Publishers Limited, 1988. — 110 p.

[6] *Kutsenko, V.* Symmetric Branching Random Walks in Random Media: Comparing Theoretical and Numerical Results / V. Kutsenko, E. Yarovaya. — URL: arXiv:2109.09126. — Загл. с титул. экрана.

[7] *Yarovaya, E.* Symmetric Branching Walks in Homogeneous and non-homogeneous Random Environments // Communications in Statistics — Simulation and Computation. — 2012. — №7. — P. 1232–1249

### Библиографическая ссылка

*Куценко, В. А.* Моделирование процессов с генерацией и транспортом частиц в случайной среде / В. А. Куценко, Е. Б. Яровая // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 190–198.
https://doi.org/10.26456/mfcsics-21-28

### Сведения об авторах

1. Владимир Александрович Куценко
Московский государственный университет им. М. В. Ломоносова. Аспирант

*Россия, 119991, ГСП-1, Москва, Ленинские горы, Главное здание, механико-математический факультет, кафедра теории вероятностей*
*E-mail: vlakutsenko@ya.ru*

2. Елена Борисовна Яровая
Московский государственный университет им. М. В. Ломоносова. Профессор

*Россия, 119991, ГСП-1, Москва, Ленинские горы, Главное здание, механико-математический факультет, кафедра теории вероятностей*
*E-mail: yarovaya@mech.math.msu.su*

УДК 519.686.4
AMS MSC2020: 68N20

# Исчисления алиасов для Си-подобных языков

**Лыгин Л. И., Шилов Н. В.**

Университет Иннополис

Аннотация. В нашей работе мы представляем реализацию двух вариантов исчисления алиасов (синонимичных адресов) для языка с адресной арифметикой — варианта, описанного в работе [2], и нового «легкого» варианта для обнаружения утечек памяти. Разработка наших вариантов исчисления была вдохновлена исчислением алиасов для языка без адресной арифметики из работы [1].

Ключевые слова: модели памяти, адресная арифметика, арифметика Пресбургера, проблема алиасинга, исчисление алиасов.

## Введение

Несмотря на то, что бóльшая часть современного программного кода написана на языках с автоматическим управлением памятью и сборкой мусора, большáя часть критического кода по-прежнему написана на процедурных языках типа Си с прямым доступом к памяти, особенно в «жестких» средах, таких как микроконтроллеры или критичные к производительности центры обработки данных. Языки типа Си позволяют программисту обращаться к памяти напрямую практически по любому адресу, потому что все указатели на память являются просто целыми числами, и всегда можно привести любое целое число к указателю. Хотя это позволяет создавать эффективные приложения, это также опасно, потому что нет никаких барьеров для предотвращения несанкционированного доступа программы (например, к нераспределенной памяти) или у т е ч е к п а м я т и (например, выделение массивов с последующим «забыванием» о них), некоторые из которых могут привести к утечкам памяти в случае сбоя во время выполнения.

В некоторых случаях могут помочь автоматические тесты, но тестирование на утечки памяти (ситуации, которые возникают, когда программа выделяет часть памяти, а затем «теряет» все ссылки на нее и поэтому не имеет возможности когда-либо ее удалить), как известно, сложно. Гораздо лучше проверять использование памяти автоматически, не выполняя целевую программу, потому что это позволяет интегрировать готовые к использованию автоматизированные инструменты в интегрированные среды разработки (IDE) и конвейеры непрерывной интеграции (CI).

Проверка использования памяти при наличии адресной арифметики (то есть в ситуации, когда программа может попытаться получить доступ к памяти в любом «вычислимом» месте памяти с целочисленным адресом) может быть решена с помощью множества методов с разной степенью точности и эффективности. Одна конкретная проблема в этой области — анализ алиасинга, определяющий, могут ли два указателя (или должны, в зависимости от подхода) указывать на одно и то же место в «куче» (то есть — динамической памяти). Заметим, что проблемы с псевдонимами (алиасами — aliases) могут возникать даже без явного использования адресной арифметики, так как многие современные языки поддерживают указатели (или обычно используют только ссылки для хранения значений), а наличие поддержки для указателей подразумевает, что могут быть две переменные, которые «указывают» на одно и то же место в памяти.

## 1. Обзор содержания выполненной работы

В нашей работе мы представляем реализацию двух вариантов одного из методов статического анализа алиасинга — исходного (описанного в работе [2]) и нового «легкого» варианта исключительно для обнаружения утечек памяти. Исходный вариант исчисления был «вдохновлен» идеей исчисления алиасов без адресной арифметики из работы [1]. Вариант исчисления алиасов из [2] вычисляет по программе на модельном процедурном языке программирования MoRe (с адресной арифметикой) наборы синонимов и антонимов (то есть равенств $x = y$ и неравенств $x \neq y$ адресных выражений), а «легкий» вариант исчисления алиасов основан исключительно на равенствах адресных выражений. Мы также приводим результаты

экспериментов по проверке корректности и эффективности обоих вариантов исчисления для обнаруживая утечки памяти в процедурных программах.

Синтаксис языка MoRe задан следующим образом:
Определение 1.

$$P ::= \; skip \mid var\ V = C \mid V := T \mid$$
$$\mid V := cons(C^*) \mid [V] := V \mid V := [V] \mid dispose(V) \mid$$
$$\mid (P; P) \mid (if\ F\ then\ P\ else\ P) \mid (while\ F\ do\ P).$$

В языке MoRe есть только два типа данных — «числа» и адреса с возможностью неявно преобразовывать числа в адреса, а числовые выражения — в адресные выражения. Про тип адресов мы предполагаем, что теория первого порядка этого типа является разрешимой. Например, если множество адресов — это все натуральные числа, тогда адресная арифметики — это арифметика Пресбургера.

В качестве примера анализа, осуществимого с использованием обоих наших вариантов исчисления алиасов для языка MoRe, рассмотрим следующую программу
Пример 1. `var x = 0 ; x := cons(1, 3) ; x := 3`

Для этой программы анализ должен обнаружить (и обнаруживает) следующие утечки памяти:

- «`Memory leak - x is no longer reachable after assignment on statement x := 3 line 3`» — адрес первого элемента «двухэлементного списка», состоящего из 1, 3, более недоступен,
- «`Memory leak - x + 1 is no longer reachable after assignment on statement x := 3 line 3`» — адрес второго элемента «двухэлементного списка», состоящего из 1, 3, более недоступен,

из-за присваивания адресной переменной `x`, хранившей адрес начала списка, нового значения.

## Заключение

Тестирование подтвердило перспективность нашего подхода для статического анализа утечек памяти, в то время как масштабируе-

мость и полезность для анализа промышленного кода нуждаются в дополнительных исследованиях.

На данный момент мы не провели сравнения нашего подхода с другими подходами к анализу алиасинга [3]. Такое сравнение поможет понять сильные и слабые стороны, а также предоставить полезную информацию о том, как можно улучшить наш анализ.

Другая важная задача — тестирование реализации на более крупных примерах. На данный момент все тестовые примеры содержат не более 20 строк кода, тестирование с использованием более крупных примеров может выявить серьезные узкие места в производительности.

Еще одна важная задача на будущее — «масштабируемость» теории, распространение подхода на языки с вызовами функций и объектно-ориентированные языки.

## Список литературы

[1] *Meyer, B.* Steps towards a theory and calculus of aliasing // International Journal of Software and Informatics. — 2011. — Vol. 5. — P. 77–116.

[2] *Shilov, N. V.* Alias calculus for a simple imperative language with decidable pointer arithmetic / N. V. Shilov, A. Satekbayeva, A. P. Vorontsov // Bulletin of the Novosibirsk Computing Center. Computer Science series. — 2014. — №37. — P. 131–148.

[3] *Smaragdakis Y.* Pointer Analysis / Y. Smaragdakis, G. Balatsouras // Foundations and Trends in Programming Languages. — 2015. — Vol. 2, №1. — P. 1–69.

## Библиографическая ссылка

*Лыгин, Л. И.* Исчисления алиасов для Си-подобных языков / Л. И. Лыгин, Н. В. Шилов // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 199–203.
https://doi.org/10.26456/mfcsics-21-29

**Сведения об авторах**

1. Лыгин Леонид Ильич
Университет Иннополис. Студент

*Россия, 420500, г. Иннополис ул. Университетская, д.1*
*E-mail: l.lygin@innopolis.ru*

2. Шилов Николай Вячеславович
Университет Иннополис. Доцент

*Россия, 420500, г. Иннополис ул. Университетская, д.1*
*E-mail: shiloviis@mail.ru*

УДК 519.248
AMS MSC2020: 60K40

# Статистический анализ случайных графов для задачи распространения информации[1]

**Маркович Н. М., Рыжов М. С.**

Институт проблем управления им. В. А. Трапезникова РАН

Аннотация. Работа посвящена распространению сообщений в случайных графах. Рассматривается задача передачи сообщения каким-то узлом графа другим узлам в графе. Для этой цели среди узлов графа находятся лидирующие узлы, которые наиболее быстро распространяют информацию, а также лидирующие сообщества, к которым такие узлы относятся. С помощью статистических методов, оценивая экстремальные и хвостовые индексы сообществ, проводится исследование фиксированных и динамически меняющихся графов, в которых распределения числа входящих и выходящих связей между узлами задается степенным законом с известными параметрами.

Ключевые слова: случайный граф, распространение сообщений, сообщество, хвостовой индекс, экстремальный индекс.

## Введение

Распространение сообщений в случайных графах является важной задачей с приложением в различных областях, как распределенные вычисления [2, 11] и социальные сети. Например, время распространения инфекции в контактных сетях [5] может влиять на эффективность вакцинации.

В прошлых работах авторов [7, 8] были исследованы распределения ПейджРангов узлов и свойства сообществ узлов посредством экстремального и хвостового индексов. В качестве сообщества рассматривалась группа узлов, которые связаны между собой большим

[1]Работа выполнена при финансовой поддержке РФФИ (грант 19-01-00090)

числом связей и мало связаны с остальными узлами графа. В настоящей работе рассматриваются результаты исследования экстремального и хвостового индексов для задачи распространения сообщений в случайном графе.

Анализируя фиксированные ненаправленные графы (Секция 2) и эволюцию во времени направленных графов (Секция 3), в статье приводятся обнаруженные зависимости между экстремальным и хвостовым индексами, оцениваемыми по множествам характеристик узлов сообществ, и временем распространения сообщения.

## 1. Основные определения

### 1.1. Экстремальный индекс

Для стационарной последовательности $\{X_n\}_{n\geqslant 1}$ с функцией распределения (ф. р.) $F(x)$ и максимумом $M_n = \max\limits_{1\leqslant j\leqslant n} X_j$ существует экстремальный индекс (ЭИ) $\theta \in [0,1]$, если для любого $0 < \tau < \infty$ найдется вещественная последовательность $u_n = u_n(\tau)$ удовлетворяющая соотношениям

$$\lim_{n\to\infty} n(1 - F(u_n)) = \tau \qquad \text{и} \qquad \lim_{n\to\infty} P\{M_n \leqslant u_n\} = e^{-\tau\theta},$$

в том смысле, что $M_n$ остается ограниченным при $n \to \infty$ ([6], p. 63).

Для независимых случайных величин ЭИ равен единице. Обратное утверждение неверно. Чем ближе $\theta$ к нулю, тем сильнее степень локальной зависимости (кластерности). Обратная величина $1/\theta$ аппроксимирует средний размер кластера, то есть среднее число превышений на кластер. В классической теории кластер может определяться как блок данных с хотя бы одним превышением уровня $u$. В [3] кластер определяется, как число $T(u)$ превышений между двумя последовательными не превышениями

$$T(u) = \min\{t \geqslant 1 : X_{j+t} > u\} \quad \text{при} \quad X_j > u. \tag{1}$$

Значение ЭИ может быть оценено с помощью интервальной оценки [3]

$$\widehat{\theta}(u) = \min(1, \theta^*), \tag{2}$$

$$\theta^* = \begin{cases} \dfrac{2\Big(\sum\limits_{i=1}^{N-1} T(u)_i - 1\Big)^2}{(N-1)\sum\limits_{i=1}^{N-1}(T(u)_i - 1)(T(u)_i - 2)}, & \max\{T(u)_i\} > 2, \\ \dfrac{2\Big(\sum\limits_{i=1}^{N-1} T(u)_i\Big)^2}{(N-1)\sum\limits_{i=1}^{N-1}(T^2(u)_i)}, & \text{иначе,} \end{cases}$$

где $N = N(u) = \sum\limits_{i=1}^{n} \mathbb{1}(X_i > u)$. Чтобы ввести интервальную оценку для графа, предлагается определить $T(u)$ как количество ребер кратчайшего пути между узлами, характеристики которых больше, чем $u$ [9]. Такое предположение помогает рассматривать ЭИ сообществ узлов, используя известные результаты, полученные для последовательностей.

### 1.2. Хвостовой индекс

Пусть $\{X_n\}_{n\geqslant 1}$ — стационарная последовательность независимых одинаково распределенных (н. о. р) случайных величин (с. в.) с ф. р. $F(x)$. Параметр $\alpha_{TI}$ называется хвостовым индексом (ХИ). Это может быть оценено с помощью оценки Хилла [4]

$$\widehat{\alpha}^H(k) = \left(\frac{1}{k}\sum_{i=1}^{k} \log\left(\frac{X_{(n-i+1)}}{X_{(n-k)}}\right)\right)^{-1}, \tag{3}$$

где $X_{(1)} \leqslant X_{(2)} \leqslant \cdots \leqslant X_{(n)}$ порядковые статистики, соответствующие выборке. $k$ — это число наибольших порядковых статистик, оптимальное значение которого выбирается методом бутстреп (bootstrap) [4].

Для исследования распределения характеристик узлов в случайном графе требуется полагать, что характеристики является

независимыми друг от друга, что в общем случае не является верным. Предполагая это условие выполненным или проверяя его с помощью статистических тестов, можно исследовать ХИ характеристик сообществ.

## 2. Распространение сообщений в ненаправленном графе

Опишем алгоритм распространения SPREAD, предложенный в [11] для неориентированного графа $G = (V, E)$. Здесь $V$ и $E$ — наборы вершин и ребер графа соответственно. Рассмотрим асинхронную модель времени, где узел может инициировать связь по тикам глобальных часов, которые моделируются как процесс Пуассона с параметром $n = |V|$, [2, 11]. Пусть $k \geqslant 0$ обозначает номер тика часов или шаг алгоритма. При каждом $k$ только один узел $i$, выбранный равномерно среди всех, может отправить все свои имеющиеся сообщения, связываясь с другим узлом $j$ с вероятностью $P_{ij} = 1/D_i$, где $D_i$ — степень или количество связей узла $i$.

Выберем узел $x$ с характеристикой $X_x$ и сообщество узлов $S$, к которому он может принадлежать. Чтобы определить ЭИ сообщества $S$ (предполагая, что ЭИ существует), берем высокий квинтиль $F(x)$ распределения выбранной характеристики в качестве порогового значения $u^*$. Для оценивания необходимо определить множество кратчайших путей $\{T(u^*)_i\}$ для всех возможных пар $(x, y) \in S$ [9]. Событие $\{T(u^*)_i = m\}$ означает, что характеристики $X_{i_1}, X_{i_2}, \ldots, X_{i_m}$ в последовательности $X_{xy}$ меньше, чем $u^*$, но $X_x$ и $X_y$ превышают $u^*$.

Моделирование показало, что узлы с высокими значениями близости (closeness centrality)

$$C_x = \frac{n-1}{\sum_{y, y \neq x} d(x, y)}, 0 < C_x \leqslant 1,$$

[12], где $d(x, y)$ — длина кратчайшего пути между узлами $x$ и $y$, распространяют информацию быстрее прочих узлов [9]. Такие узлы являются лидирующими узлами для задачи распространения информации. Также верно, что узлы с большим количеством связей $D_x$ быстрее распространяют сообщения. Также были определены

лидирующие сообщества, содержащие наилучших узлов распространителей. Такие сообщества имеют такие же значения ЭИ, что и весь граф, рассмотренный как отдельное сообщество.

## 3. Распространение сообщений в направленном динамически меняющемся графе

Метод эволюции линейного предпочтительного присоединения (ПА) (Preferential Attachment) [1,13] начинает работу с начального ориентированного графа $G(k_0)$ с хотя бы одним узлом и $k_0$ ребрами. Для неотрицательных параметров $\alpha, \beta, \gamma$, таких как $\alpha+\beta+\gamma=1$, и $\Delta_{in}, \Delta_{out}$, ПА строит растущую последовательность направленных случайных графов $G(k) = (V(k), E(k))$. Граф $G(k)$ создается из $G(k-1)$ путем добавления нового направленного ребра. Обозначим число узлов на шаге $k$ через $N(k)$, а число входящих (in-degree) и выходящих связей (out-degree) узла $w$ в графе $G(k)$ с ребрами $k$ как $I_k(w)$ и $O_k(w)$. В [1,13] предложены три сценария создания ребра. На каждом шаге алгоритма путем подбрасывания 3-сторонней монеты с вероятностями $\alpha$, $\beta$ и $\gamma$ выбирается один из сценариев:

- В соответствии с $\alpha$-схемой добавляется новый узел $w_{new}$ и ребро $(w_{new} \to w)$ с вероятностью $\alpha$. Существующий узел $w \in V(k-1)$ выбирается с вероятностью

$$P(w \in V(k-1)) = \frac{I_{k-1}(w) + \Delta_{in}}{k-1+\Delta_{in}N(k-1)}.$$

- В соответствии с $\beta$-схемой добавляется новое ребро $(w_1 \to w_2)$ с вероятностью $\beta$, где оба существующих узла $w_1$ и $w_2$ выбираются независимо и с вероятностью

$$P(w_1 \to w_2) = \frac{O_{k-1}(w_1) + \Delta_{out}}{k-1+\Delta_{out}N(k-1)} \cdot \frac{I_{k-1}(w_2) + \Delta_{in}}{k-1+\Delta_{in}N(k-1)}.$$

- В соответствии с $\gamma$-схемой добавляется новый узел $w_{new}$ и ребро $(w \to w_{new})$ с вероятностью $\gamma$, $w \in V(k-1)$ выбирается с вероятностью

$$P(w \in V(k-1)) = \frac{O_{k-1}(w) + \Delta_{out}}{k-1+\Delta_{out}N(k-1)}.$$

Это означает, что $N(k) = N(k-1)$ для $\beta$-схемы и $N(k) = N(k-1)+1$ для остальных.

Несмотря на то, что линейный ПА используется для эволюции ориентированных графов, он так может быть использован как модель для распространения информации [10]. Предполагая, что сообщение, находящееся в одном из узлов, распространяется среди фиксированного числа $n$ узлов, на каждом шаге ПА с предопределенными значениями параметров сообщение может быть доставлено от узла $i$ в узел $j$ только, если создано направленное ребро $(i \to j)$. Такое ребро может быть добавлено к сети только с помощью $\gamma-$ или $\beta-$ схем. Если узел $i$ не имеет сообщения, то ребро $(i \to j)$ не распространяет сообщение дальше на узел $j$. Схема $\alpha$ увеличивает число узлов без сообщения.

Для анализа скорости распространения сообщений с помощью ПА модели было проведено ее сравнение с алгоритмом SPREAD при $P_{ij} = 1/O_i$ и $P_{ij} = 1/(O_i + I_i)$. Здесь $P_{ij}$ обозначает вероятность, что узел $i$ выберет узел $j$ для передачи сообщения. При исследовании некоторых модельных графов было показано, что ПА может быстрее распространять информацию для некоторых наборов параметров $(\alpha, \beta, \gamma)$ при $\Delta_{in} = \Delta_{out} = 1$, чем SPREAD [10].

Также было проведено моделирование неоднородных графов, состоящих из сообществ узлов с разными распределениями числа входящих и выходящих связей. Эти распределения, как показано многими авторами, имеют правильно меняющиеся хвосты. Было обнаружено, что узлы из сообществ с наименьшим ХИ для распределения out-degree, то есть с распределением, имеющим наиболее тяжелый хвост, распространяют свое сообщение быстрее, чем узлы из прочих сообществ [10]. Такие сообщества могут быть названы лидирующими.

## Заключение

В работе исследовались фиксированные ненаправленные графы (Секция 2) и динамически развивающиеся направленные графы (Секция 3). Используя непараметрические оценки экстремального и хвостового индексов сообщества узлов графа, было обнаружено существование лидирующих сообществ, то есть групп, связанных между собой большим числом ребер и содержащих узлы — лидеры

по скорости распространения информации. Моделирование на ряде примеров графов в работах [9] и [10] показало следующие результаты. Для фиксированных ненаправленных графов лидирующие сообщества имеют то же значение экстремального индекса, что и весь граф. Для динамических направленных графов, эволюционирующих во времени, лидирующие сообщества обладают наименьшим хвостовым индексом для числа выходящих связей (out-degree) по сравнению с прочими сообществами.

**Список литературы**

[1] *Bollobás, B.* Directed scale-free graphs / B. Bollobás, C. Borgs, J. Chayes, O. Riordan // In Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms (SODA '03). — Philadelphia, Pennsylvania : Society for Industrial and Applied Mathematics, 2003. — P. 132–139.

[2] *Censor-Hillel, K.* Partial Information Spreading with Application to Distributed Maximum Coverage / K. Censor-Hillel, H. Shachnai // In Proceedings of the 29th ACM symposium on Principles of distributed computing (PODC '10). — New York, N. Y.: ACM, 2010. — P. 161–170.

[3] *Ferro, C.* Inference for clusters of extreme values /C. Ferro, J. Segers // Journal of the Royal Statistical Society. Series B (Statistical Methodology). — 2003. — Vol. 65, №2. — P. 545–556.

[4] *Hall, P.* Using the Bootstrap to Estimate Mean Squared Error and Select Smoothing Parameter in Nonparametric Problems // Journal of Multivariatte Analysis. — 1990. — Vol. 32. — P. 177–203.

[5] *Holme, P.* Cost-efficient vaccination protocols for network epidemiology / P. Holme, N. Litvak // PLoS Computational Biology. — 2017. — Vol. 13, №9. — e1005696.

[6] *Leadbetter, M. R.* Extremes and local dependence in stationary sequences // Zeitschrift fur Wahrscheinlichkeitstheorie und Verwandte Gebiete. — 1983. — Vol. 65. — P. 291–306.

[7] *Markovich, N. M.* Nonparametric Analysis of Extremes on Web Graphs: PageRank versus Max-Linear Model / N. M. Markovich, M. S. Ryzhov, U. R. Krieger // CCIS-2017. — Vol. 700. — P. 13–26.

[8] *Markovich, N. M.* Statistical Clustering of a Random Network by Extremal Properties / N. M. Markovich, M. S. Ryzhov, U. R. Krieger // CCIS-2018. — Vol. 919. — P. 71–82.

[9] *Markovich, N. M.* Leader Nodes in Communities for Information Spreading / N. M. Markovich, M. S. Ryzhov // LNCS. — 2020. — Vol. 12563. — P. 475–484.

[10] *Markovich, N. M.* Information Spreading with Application to Non-homogeneous Evolving Networks / N. M. Markovich, M. S. Ryzhov // DCCN 2021 (Принято к публикации).

[11] *Mosk-Aoyama, D.* Computing separable functions via gossip / D. Mosk-Aoyama, D. Shah // In Proceedings of the 25th ACM symposium on Principles of distributed computing (PODC '06). — New York, N. Y.: ACM, 2006. — P. 113–122.

[12] *Newman, M. E. J.* Networks: An Introduction. — 2nd ed. — Oxford : Oxford University Press, 2018. — 800 p.

[13] *Wan P.* Are extreme value estimation methods useful for network data? / P. Wan, T. Wang, R. A. Davis, S. I. Resnick // Extremes. — 2020. — Vol. 23. — P. 171–195.

### Библиографическая ссылка

*Маркович, Н. М.* Статистический анализ случайных графов для задачи распространения информации / Н. М. Маркович, М. С. Рыжов // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 204–212.
https://doi.org/10.26456/mfcsics-21-30

### Сведения об авторах

1. Наталья Михайловна Маркович
Институт проблем управления им. В. А. Трапезникова РАН. Главный научный сотрудник

*Россия, 117997, Москва, ул. Профсоюзная, 65*
*E-mail: markovic@ipu.rssi.ru , nat.markovich@gmail.com*

2. Максим Сергеевич Рыжов
Институт проблем управления им. В. А. Трапезникова РАН. Младший научный сотрудник

*Россия, 117997, Москва, ул. Профсоюзная, 65*
*E-mail: maksim.ryzhov@frtk.ru*

УДК 519.765
AMS MSC2020: 68Q45

# Верификация криптографических протоколов[1]

**Миронов А. М.**

Университет Иннополис;
Лидирующий Исследовательский Центр

Аннотация. В работе излагается новая математическая модель криптографических протоколов, и приводится пример применения этой модели для верификации протоколов аутентификации.

Ключевые слова: криптографический протокол, процессная модель, верификация.

## Введение

### Понятие криптографического протокола

Криптографический протокол (КП) представляет собой распределенный алгоритм, описывающий порядок обмена сообщениями между несколькими агентами. Примеры таких агентов — компьютерные системы, банковские карточки, люди, и т. д.

Для обеспечения свойств безопасности КП (таких например как конфиденциальность передаваемых данных) в КП могут использоваться криптографические преобразования (шифрование, электронная подпись, хэш-функции, и т. п.). Мы предполагаем, что криптографические преобразования, используемые в КП, являются идеальными, то есть удовлетворяют некоторым аксиомам, выражающим, например, невозможность извлечения открытых текстов из шифртекстов без знания соответствующих криптографических ключей.

[1]Исследование выполнено при финансовой поддержке Министерства цифрового развития, связи и массовых коммуникаций РФ и АО «Российская венчурная компания» (договор №004/20 от 20.03.2020, ИГК 0000000007119P190002)

**Уязвимости в криптографических протоколах**

Многие уязвимости в КП связаны не с плохими криптографическими качествами используемых в них криптографических примитивов, а с логическими ошибками в КП. Наиболее ярким примером уязвимости в КП является уязвимость в КП аутентификации Нидхэма – Шредера [3], который был опубликован в 1978 г., и использовался в критических по безопасности информационных системах. Спустя более 16 лет после начала использования этого КП в нем обнаружилась логическая ошибка [2], связанная с возможностью непредусмотренного нечестного поведения одного из участников этого КП и подрывающая безопасность этого КП. Особенность этой ошибки заключается в том, что данный КП является предельно простым распределенным алгоритмом, состоящим всего из трех действий, и при визуальном анализе этого КП отсутствие в нем ошибок не вызывало никаких сомнений. Ошибка была обнаружена лишь при помощи инструмента автоматизированной верификации КП.

Другой пример логической ошибки в КП [1]: в КП входа в портал Google, позволяющем пользователю идентифицировать себя только один раз, а затем обращаться к различным приложениям (таким, например, как Gmail или календарь Google), обнаружена логическая ошибка, позволяющая нечестному поставщику услуг выдавать себя за любого из своих пользователей для другого поставщика услуг.

Также есть примеры уязвимостей в КП, используемых для аутентификации перед провайдерами мобильной телефонной связи, для снятия денег в банкомате, для работы с электронными паспортами, проведения электронных выборов, и т. д.

Все эти примеры являются обоснованием того, что в критических по безопасности системах недостаточно неформального анализа требуемых свойств безопасности используемых в них КП, необходимо

- построение математических моделей анализируемых КП,
- описание свойств анализируемых КП в виде математических объектов, называемых спецификациями свойств этих КП, и
- построение формальных доказательств утверждений о том, что анализируемые КП удовлетворяют (или не удовлетворяют)

своим спецификациям, процедура построения таких доказательств называется верификацией анализируемых КП.

В настоящей работе строится новая математическая модель КП, в терминах которой можно выражать такие свойства корректности КП, как например целостность и конфиденциальность передаваемых сообщений (то есть обоснование следующих свойств анализируемого КП: сообщения, посланные одним участником этого КП другому участнику этого КП, доходят до получателя в неискаженном виде, и содержание этих сообщений не будет известно противнику), или аутентификация (то есть доказательство подлинности) участников КП.

## 1. Последовательные и распределенные процессы

В этом параграфе мы излагаем понятия последовательного и распределенного процессов. Последовательный процесс является моделью участника КП, а распределенный процесс является моделью всего КП.

### 1.1. Вспомогательные понятия

#### 1.1.1. Термы

Предполагаем, что заданы множества $Types$, $Con$, $Var$ и $Fun$, элементы которых называются типами, константами, переменными, и функциональными символами (ФС), соответственно.

Каждому элементу $x$ множеств $Con$, $Var$ и $Fun$ сопоставлен некоторый тип $\tau(x) \in Types$, причем если $x \in Fun$, то $\tau(x)$ имеет вид $(\tau_1, \ldots, \tau_n) \to \tau$, где $\tau_1, \ldots, \tau_n, \tau \in Types$.

Ниже определяется множество $Tm$ термов, которые предназначены для описания сообщений, пересылаемых во время выполнения КП. Множество $Tm$ определяется индуктивно. Каждому терму $e$ сопоставлен некоторый тип $\tau(e) \in Types$. Определение терма имеет следующий вид:

- $x$ является термом типа $\tau(x)$ для любого $x \in Con \cup Var$,

- если $f \in Fun$, $e_1, \ldots, e_n$ — термы, и $\tau(f) = (\tau(e_1), \ldots, \tau(e_n)) \to \tau$, то запись $f(e_1, \ldots, e_n)$ является термом типа $\tau$.

Будем использовать следующие обозначения:

- $Var(e) = \{x \in Var \mid x \text{ входит в } e\}$ для любого $e \in Tm$,
- $Tm(X) = \{e \in Tm \mid Var(e) \subseteq X\}$ для любого $X \subseteq Var$,
- $E_{\mathbf{X}} = E \cap Var$ для любого $E \subseteq Tm$, $E_\tau = \{e \in E \mid \tau(e) = \tau\}$ для любого $\tau \in Types$.

### 1.1.2. Примеры типов

Будем считать, что $Types$ содержит следующие типы:

- тип $\mathbf{A}$, термы этого типа называются агентами,
- тип $\mathbf{C}$, термы этого типа называются каналами, они обозначают каналы связи, при помощи которых агенты взаимодействуют друг с другом путем передачи сообщений,
- тип $\mathbf{K}$, термы этого типа называются ключами, они обозначают криптографические ключи, которые агенты могут использовать для шифрования или дешифрования сообщений,
- тип $\mathbf{M}$, термы этого типа называются сообщениями, они обозначают сообщения, которые агенты могут пересылать друг другу во время своей работы,
- тип $\mathbf{N}$, термы этого типа называются нонсами, они обозначают переменные с уникальными значениями,
- тип $\mathbf{P}$, термы этого типа называются процессами.

Записи *Agents*, *Channels*, *Keys*, *Messages*, *Nonces* и *Processes* обозначают множества всех агентов, каналов, ключей, сообщений, нонсов и процессов, соответственно.

Будем использовать следующие соглашения и обозначения:

- множество *Channels* содержит переменную, обозначаемую символом $\circ$, и называемую открытым каналом,

- тип $\mathbf{M}$ включает все другие типы из $Types$, то есть терм любого типа является также термом типа $\mathbf{M}$,
- для любых $n \geqslant 1, \tau \in Types$ множество $Types$ содержит тип $\tau_n$, значения которого — кортежи длины $n$ из значений типа $\tau$.

### 1.1.3. Примеры функциональных символов

Будем предполагать, что $Fun$ содержит следующие ФС.

- ФС $tuple_n$, где $n \geqslant 1$ и $\tau(tuple_n) = (\underbrace{\mathbf{M}, \ldots, \mathbf{M}}_{n}) \to \mathbf{M}_n$.

  Для каждого списка $(e_1, \ldots, e_n)$ термов терм $tuple_n(e_1, \ldots, e_n)$ будет обозначаться более короткой записью $(e_1, \ldots, e_n)$.

- ФС $pr_{n,i}$, где $n \geqslant 1$, $i \in \{1, \ldots, n\}$, и $\tau(pr_{n,i}) = \mathbf{M}_n \to \mathbf{M}$.

  Для любого $e \in Tm_{\mathbf{M}_n}$ терм $pr_{n,i}(e)$ является $i$-й компонентой кортежа $e$, и будет обозначаться записью $(e)_i$.

- ФС $hash\_function$ (возможно с индексами) типа $\mathbf{M} \to \mathbf{M}$.

  Терм $hash\_function(e)$ обозначает значение хэш-функции сообщения $e$.

- ФС $encrypt$ и $decrypt$ типа $(\mathbf{K}, \mathbf{M}) \to \mathbf{M}$.

  Термы вида $encrypt(k, e)$ и $decrypt(k, e)$ обозначают сообщения, получаемые шифрованием (и дешифрованием, соответственно) сообщения $e$ на ключе $k$. Термы вида $encrypt(k, e)$ будут обозначаться записями $k(e)$, данные термы называются шифрованными сообщениями (ШС).

- ФС $shared\_key$ типа $\mathbf{A}_n \to \mathbf{K}$, где $n \geqslant 2$ (то есть одно и то же обозначение $shared\_key$ используется для семейства ФС).

  Терм вида $shared\_key(A_1, \ldots, A_n)$ называется разделяемым ключом агентов $A_1, \ldots, A_n$ и будет обозначаться записью $k_{A_1 \ldots A_n}$.

- ФС $digital\_signature$ типа $(\mathbf{M}, \mathbf{A}) \to \mathbf{M}$.

  Терм вида $digital\_signature(e, A)$ обозначает цифровую подпись сообщения $e$, сделанную агентом $A$.

  Тройка $(e, A, digital\_signature(e, A))$ будет обозначаться $(e)_A$.

Будем использовать следующие обозначения: для любого $e \in Tm$

$$VarEncKeys(e) \quad = \{k \in Var_{\mathbf{K}} \mid \exists\, e' \in Tm : k(e') \subseteq e\}.$$

### 1.1.4. Выражения

В этом пункте определяется множество $Expr$ выражений, которые предназначены для описания множеств термов. Например, в качестве такого множества может выступать совокупность термов, доступных в текущий момент какому-либо процессу, или совокупность сообщений, находящихся в текущий момент в каком-либо канале.

Выражением называется запись одного из следующих видов:

- $E$, где $E \subseteq Tm$,
- $[P]$ и $[c]$, где $P \in Processes$, $c \in Channels$,
- $k^{-1}(E)$, где $k \in Keys$, и $E \in Expr$,
- $E \cap E'$, $E \cup E'$, $\neg E$, где $E, E' \in Expr$.

$Var(E) = \{x \in Var \mid x$ входит в $E\}$ для любого $E \in Expr$.

Выражения вида $k^{-1}([P])$ и $k^{-1}([c])$ обозначаются $k^{-1}[P]$ и $k^{-1}[c]$ соответственно. Выражения вида $\{e\}$, где $e \in Tm$, обозначаются без фигурных скобок. Ниже каждому выражению сопоставляется значение этого выражения в текущий момент времени, которое является множеством термов.

### 1.1.5. Формулы

В этом пункте определяется понятие формулы, которое предназначено для описания свойств множеств термов. В определении данного понятия используется понятие элементарной формулы (ЭФ), которая представляет собой запись одного из следущих видов:

1) $e \in E$, $E = E'$, $E \subseteq E'$, $E \supseteq E'$, где $e \in Tm$, $E, E' \in Expr$,
2) $E \perp_{\mathbf{C}} P$ и $E \perp_{\mathbf{K}} P$, где $E \subseteq Tm, P \in Processes$,
3) $at_P = i$, где $P \in Processes$.

ЭФ выражают свойства значений входящих в них выражений в текущий момент времени. ЭФ из первого пункта выражают свойства, соответствующие входящим в них теоретико-множественным символам. ЭФ из второго пункта выражают свойства, изложенные в пункте 1.2.4, ЭФ из третьего пункта выражают свойства текущего состояния последовательного процесса, подробнее см. в пункте 1.2.3.

Ф о р м у л о й называется произвольная совокупность ЭФ. Каждая формула $\varphi = \{\varphi_i \mid i \in I\}$ выражает утверждение, представляющее собой конъюнкцию утверждений, выражаемых ЭФ $\varphi_i$ $(i \in I)$.

Множество всех формул обозначается записью $Fm$. Для любого $\varphi \in Fm$ запись $Var(\varphi)$ обозначает множество переменных, входящих в $\varphi$.

Для любого списка формул $\varphi_1, \ldots, \varphi_n \in Fm$ формула $\varphi_1 \cup \ldots \cup \varphi_n$ будет обозначаться записью $\{\varphi_1, \ldots, \varphi_n\}$.

### 1.1.6. Связывания

С в я з ы в а н и е — это функция $\theta : Var \to Tm$. Будем говорить, что связывание $\theta$ связывает переменную $x \in Var$ с термом $\theta(x)$.

Будем использовать следующие обозначения:

- множество всех связываний обозначается символом $\Theta$,
- $id$ обозначает тождественное связывание: $id(x) = x$ для любого $x \in Var$,
- $\Theta(X) = \{\theta \in \Theta \mid \forall\, x \in Var \setminus X \;\; \theta(x) = x\}$ для любого $X \subseteq Var$,
- связывание $\theta \in \Theta$ может обозначаться записями

$$x \mapsto \theta(x) \quad \text{или} \quad (\theta(x_1)/x_1, \ldots, \theta(x_n)/x_n), \tag{1}$$

  вторая запись в (1) используется, когда $\theta \in \Theta(\{x_1, \ldots, x_n\})$,
- для любых $\theta \in \Theta, e \in Tm$ запись $e^\theta$ обозначает терм, получаемый из $e$ заменой для всех $x \in Var(e)$ каждого вхождения $x$ в $e$ на терм $\theta(x)$, терм $e$ называется ш а б л о н о м терма $e^\theta$ относительно $\theta$,
- для любых $\theta \in \Theta, E \subseteq Tm$ запись $E^\theta$ обозначает множество $\{e^\theta \mid e \in E\}$,

- для всех $\theta, \theta' \in \Theta$ запись $\theta\theta'$ обозначает связывание $x \mapsto (x^\theta)^{\theta'}$.

Пусть $X \subseteq X' \subseteq Var$, $\theta \in \Theta(X)$, $\theta' \in \Theta(X')$. $\theta'$ называется продолжением $\theta$, если $\theta(x) = \theta'(x)$ для любого $x \in X$.

## 1.2. Последовательные процессы

### 1.2.1. Действия

Действие — это запись одного из следующих видов:

$$c!e, \quad c?e, \quad e := e', \quad \text{где } c \in Channels,\ e, e' \in Tm,$$

которые называются посылкой сообщения $e$ в канал $c$, приемом сообщения $e$ из канала $c$, и присваиванием, соответственно.

Множество всех действий обозначается записью $Act$. Для любого $\alpha \in Act$ множество всех переменных, входящих в $\alpha$, обозначается записью $Var(\alpha)$.

Если $\theta \in \Theta$ и $\alpha \in Act$, то запись $\alpha^\theta$ обозначает действие $c^\theta!e^\theta$, $c^\theta?e^\theta$ и $e^\theta := (e')^\theta$, если $\alpha = c!e$, $c?e$ и $e := e'$, соответственно.

### 1.2.2. Понятие последовательного процесса

Последовательный процесс (ПП) — это четверка $(P, A, X, \bar{X})$, компоненты которой имеют следующий смысл:

- $P$ — граф с выделенной вершиной (называемой начальной вершиной, и обозначаемой записью $Init(P)$), каждому ребру которого сопоставлена метка $\alpha \in Act$,
- $A$ — агент, связанный с этим ПП,
- $X \subseteq Var$ — инициализированные переменные,
- $\bar{X} \subseteq X$ — скрытые переменные, они обозначают секретные ключи, скрытые каналы, или нонсы, эти переменные инициализированы уникальными значениями.

ПП является формальным описанием поведения динамической системы, работа которой заключается в последовательном выполнении действий, связанных с посылкой или приемом сообщений, а также с инициализацией неинициализированных переменных.

Для каждого ПП $(P, A, X, \bar{X})$

- данный ПП может сокращенно обозначаться тем же символом $P$, что и соответствующий ему граф, множество вершин графа $P$ также обозначается символом $P$,
- $Agent(P)$, $X(P)$, $\bar{X}(P)$ обозначают соответствующие компоненты $P$, $Var(P)$ обозначает множество всех переменных, входящих в $P$,
- $\tilde{X}(P)$ обозначает множество $X(P) \setminus \bar{X}(P)$ инициализированных нескрытых переменных процесса $P$,
- $\hat{X}(P)$ обозначает множество $Var(P) \setminus X(P)$ неинициализированных переменных процесса $P$.

С каждым ПП связана переменная из множества *Processes*, называемая именем этого ПП. Будем обозначать имена ПП теми же записями, которыми обозначаются сами ПП.

Действия вида $\circ!e$ и $\circ?e$ будут более коротко обозначаться записями $!e$ и $?e$ соответственно.

### 1.2.3. Состояние последовательного процесса

Состояние ПП $P$ — это пятерка

$$s = (at, \alpha, [P], \theta, \{[c] \mid c \in Channels\})$$

где

- $at \in P$ — вершина графа $P$ в состоянии $s$,
- $\alpha \in \{init\} \sqcup Act$ — действие перед переходом в $s$,
- $[P] \subseteq Var$ — множество инициализированных переменных в $s$,
- $\theta \in \Theta([P])$ — связывание в $s$,

- для любого $c \in \mathit{Channels}$ $[c] \subseteq Tm$ — содержимое канала $c$ в $s$.

Компоненты состояния $s$ обозначаются записями $at_s$, $\alpha_s$, $[P]_s$, $\theta_s$, $[c]_s$ соответственно. Будем обозначать записью $\langle P\rangle_s$ множество $Tm([P]_s)$.

Состояние ПП $P$ называется начальным (и обозначается $0_P$), если оно имеет вид $(\mathit{Init}(P), \mathit{init}, X(P), \mathit{id}, \{\varnothing \mid c \in \mathit{Channels}\})$.

### 1.2.4. Значения выражений и формул в состояниях последовательных процессов

Пусть заданы ПП $P$, состояние $s$, выражение $E$, и формула $\varphi$.

Запись $E^s$ обозначает множество термов, называемое значением $E$ в $s$, и определяемое следующим образом:

- $E^s = \{e^{\theta_s} \mid e \in E\}$ для любого $E \subseteq Tm$,
  для любого $e \in Tm$ множество вида $\{e\}^s$, а также единственный элемент этого множества, будем обозначать записью $e^s$,
- $[P]^s = ([P]_s)^s$, $\langle P\rangle^s = (\langle P\rangle_s)^s$, $[c]^s = [c^s]_s$, где $P \in \mathit{Processes}$, $c \in \mathit{Channels}$,
- $k^{-1}(E)^s = \{e \in Tm \mid \exists\, e' \in E^s : k^s(e) \subseteq e'\}$,
- $(E \cap E')^s = E^s \cap (E')^s$, $(E \cup E')^s = E^s \cup (E')^s$, $(\neg E)^s = Tm \setminus E^s$.

Запись $s \models \varphi$ обозначает утверждение «$\varphi$ истинна в $s$». Это утверждение верно, если $Var(\varphi)_{\mathbf{P}} \subseteq \{P\}$, и выполнено одно из условий:

- $\varphi = (e \in E)$, $(E = E')$, $(E \subseteq E')$, или $(E \supseteq E')$, где $e \in Tm$, $E, E' \in \mathit{Expr}$, и

  $e^s \in E^s$, $E^s = (E')^s$, $E^s \subseteq (E')^s$, $E^s \supseteq (E')^s$, соответственно

- $\varphi = (E \perp_{\mathbf{C}} P)$, $\mathit{Agent}(P) \notin e$ для любого $e \in E^s$ и

$$\left.\begin{array}{l} \forall\, x \in E^s_{\mathbf{X}}, \forall\, y \in [P]_s \;\; x \notin y^s \\ \forall\, x \in E^s_{\mathbf{X}}, \forall\, c \in \mathit{Channels}, \\ \qquad\qquad \text{если } \exists\, e \in [c]_s : x \in e, \text{ то } c \in E^s \end{array}\right\} \qquad (2)$$

(2) можно интерпретировать как следующее утверждение: каждая переменная из $E^s_{\mathbf{X}}$ не входит в термы, доступные процессу $P$ в состоянии $s$, и входит в термы из содержимого только таких каналов, которые недоступны для $P$,

- $\varphi = (E \perp_{\mathbf{K}} P)$, $Agent(P) \notin e$ для любого $e \in E^s$ и

$$\left.\begin{array}{l} \forall\, x \in E^s_{\mathbf{X}},\, \forall\, y \in [P]_s \;\; x \perp_{\mathbf{K},E} y^s \\ \forall\, x \in E^s_{\mathbf{X}},\, \forall\, c \in Channels, \forall\, e \in [c]_s \;\; x \perp_{\mathbf{K},E} e \end{array}\right\} \qquad (3)$$

где $x \perp_{\mathbf{K},E} e$, означает, что

$$\begin{array}{l} \text{каждое вхождение } x \text{ в } e \text{ содержится} \\ \text{в подтерме } k(\ldots) \subseteq e, \text{ где } k \in E^s_{\mathbf{K}} \end{array} \qquad (4)$$

(3) можно интерпретировать как следующее утверждение: переменные из $E^s_{\mathbf{X}}$ входят в термы, доступные процессу $P$ в состоянии $s$, а также в термы из содержимого произвольного канала, в «защищенном» виде, то есть входят в подтермы вида $k(\ldots)$, где $k \in E^s_{\mathbf{K}}$,

- $\varphi = (at_P = i)$, и $at_s = i$,
- $\varphi = \{\varphi_i \mid i \in I\}$ — совокупность ЭФ, и $s \models \varphi_i$ для любого $i \in I$.

### 1.2.5. Выполнение последовательного процесса

Выполнение ПП $P$ можно понимать как обход вершин $P$, начиная с $Init(P)$, с выполнением действий, являющихся метками проходимых ребер. С каждым шагом выполнения ПП $P$ связано некоторое состояние $s$ ПП $P$, называемое текущим состоянием ПП $P$ на этом шаге (на первом шаге текущим состоянием является $0_P$). Если текущий шаг выполнения ПП $P$ не является заключительным, то на этом шаге происходит замена текущего состояния $s$ на состояние $s'$, которое будет текущим состоянием на следующем шаге, для этого

1) либо выбирается выходящее из $at_s$ ребро графа $P$, метка $\alpha$ которого обладает следующими свойствами:

- если $\alpha^{\theta_s}$ содержит вхождение терма вида $shared\_key(\ldots)$, то $Agent(P)$ присутствует в этом вхождении,
- выполнено одно из условий:

$$\left.\begin{array}{ll} \text{(а)} & \alpha = c!e,\ c, e \in \langle P\rangle_s \\ \text{(б)} & \alpha = c?e,\ c \in \langle P\rangle_s, \\ & \quad VarEncKeys(e^s) \subseteq [P]_s, \\ & \quad \exists\, \theta \in \Theta(\, Var(e) \setminus [P]_s) : (e^\theta)^s \in [c]^s \\ \text{(в)} & \alpha = (e := e'),\ e' \in \langle P\rangle_s, \\ & \quad VarEncKeys(e^s) \subseteq [P]_s, \\ & \quad \exists\, \theta \in \Theta(\, Var(e) \setminus [P]_s) : e^\theta = e' \end{array}\right\} \qquad (5)$$

и компоненты состояния $s'$ имеют следующий вид: $at_{s'}$ — конец выбранного ребра, $\alpha_{s'} = \alpha$, и

- если верно (а) в (5), то $[P]_{s'} = [P]_s$, $\theta_{s'} = \theta_s$, $[c^s]_{s'} = [c^s]_s \cup \{e^s\}$, $[c']_{s'} = [c']_s$ для любого $c' \in Channels \setminus \{c^s\}$,
- если верно (б) или (в) в (5), то $[P]_{s'} = [P]_s \cup Var(e)$, $\theta_{s'} = \theta\theta_s$, $[c']_{s'} = [c']_s$ для любого $c' \in Channels$, (будем говорить, что при переходе от $s$ к $s'$ каждая переменная $x \in Var(e) \setminus [P]_s$ инициализируется значением $x^{\theta_{s'}}$, которое становится доступным $P$),

2) либо все компоненты состояния $s'$, кроме последней, совпадают с соответствующими компонентами состояния $s$, и для любого $c \in Channels$ множество $[c]_{s'}$ либо совпадает с $[c]_s$, либо получается путем добавления терма к множеству $[c]_s$ в результате выполнения текущего шага другим ПП.

Если имеет место первая (вторая) из указанных выше ситуаций, то будем говорить, что $s'$ получается а к т и в н ы м (соответственно, п а с с и в н ы м) переходом из $s$. Запись $s \xrightarrow{P} s'$ ($s \to s'$) обозначает, что $s'$ получается активным (соответственно, пассивным) переходом из $s$.

Во время каждого выполнения каждого ПП $P$ переменные из $Var(P)$ имеют следующие особенности: для любого $x \in Var(P)$

1) если $x \in \hat{X}(P)$, то в начальный момент каждого выполнения ПП $P$ переменная $x$ не инициализирована, то есть ей не сопоставлено никакого значения,

2) если $x \in \bar{X}(P)$, то это означает, что в начальный момент каждого выполнения $Exec$ ПП $P$ данная переменная инициализирована уникальным значением, то есть значением, которое отличается от значений, сопоставленных другим инициализированным переменным при выполнении $Exec$, и от значений, сопоставленных инициализированным переменным при любом выполнении $Exec' \neq Exec$ любого ПП.

Интерпретация условий, описанных в (5), имеет следующий вид.

- Условие в (а) связано с выполнением посылки сообщения: имя $c^s$ канала, в который посылается сообщение, должно быть доступно процессу $P$ в состоянии $s$, и посылаемое сообщение $e^s$ должно быть термом, компоненты которого также доступны процессу $P$ в состоянии $s$.

- Условие в пункте (б) связано с выполнением приема сообщения:

  – имя $c^s$ канала, из которого принимается сообщение, должно быть доступно процессу $P$ в состоянии $s$,

  – все ШС в принимаемом сообщении, которые дешифруются во время приема этого сообщения, и зашифрованы не на разделяемом ключе, имеют вид $k(\ldots)$, где значение ключа $k$ должно быть доступно процессу $P$ в состоянии $s$, это свойство выражается во второй строке пункта (б) в (5),

  – терм $e$ является шаблоном некоторого терма из $[c]^s$ относительно некоторого продолжения связывания $\theta_s$, данное свойство выражается в последней строке пункта (б) в (5).

- Условие в пункте (с) связано с выполнением присваивания:

  – каждая компонента $(e')^s$ должна быть доступна $P$ в $s$,

– смысл свойств во второй и третьей строках пункта (c) в (5) совпадает со смыслом соответствующих свойств в пункте (b): каждое ШС в $(e')^s$, которое должно быть дешифровано во время выполнения этого присваивания, должно иметь вид $k(\ldots)$, причем

  * либо $k$ разделяемый ключ,

  * либо $k \in Var_{\mathbf{K}}$ и значение ключа $k$ должно быть доступно $P$ в состоянии $s$,

– терм $e$ является шаблоном терма $e'$ относительно некоторого связывания $\theta \in \Theta(\mathit{Var}(e) \setminus [P]_s)$.

### 1.2.6. Процесс противника

Процесс противника — это ПП, обозначаемый записью $P_\dagger$, и обладающий следующими свойствами:

- граф ПП $P_\dagger$ состоит из единственной вершины,
- для любого $\tau \in \mathit{Types}$ множества $\bar{X}(P_\dagger)_\tau$ и $\hat{X}(P_\dagger)_\tau$ счетны,
- для любого $\alpha \in \mathit{Act}$ граф $P_\dagger$ содержит ребро с меткой $\alpha$.

Ниже будем предполагать, что $P_\dagger$ — единственный из всех рассматриваемых ПП, граф которого имеет циклы.

### 1.2.7. Переименование переменных

Переименование переменных (называемое также просто переименованием) — это инъективная функция $\eta : X \to X'$, где $X, X' \subseteq \mathit{Var}$. Для каждого переименования $\eta : X \to X'$, каждого $e \in \mathit{Tm}$ и каждого ПП $P$ записи $e^\eta$ и $P^\eta$ обозначают терм или ПП соответственно, получаемые из $e$ или $P$ заменой для любого $x \in X$ каждого вхождения $x$ на $\eta(x)$.

Если переименование $\eta$ имеет вид $\eta : \bar{X}(P) \cup \hat{X}(P) \to \mathit{Var} \setminus \tilde{X}(P)$, то ПП $P$ и $P^\eta$ будем рассматривать как равные.

### 1.3. Распределенные процессы

В этом пункте вводится понятие распределенного процесса, которое является моделью КП. Все КП, рассматриваемые в этом тексте, мы будем отождествлять с соответствующими им распределенными процессами.

#### 1.3.1. Понятие распределенного процесса

Р а с п р е д е л е н н ы й  п р о ц е с с  (РП) — это семейство ПП:

$$\mathcal{P} = \{P_i \mid i \in I\}$$

(некоторые из которых могут совпадать). С каждым РП связана переменная типа **P**, называемая и м е н е м этого РП.

РП $\mathcal{P}$ является моделью распределенного алгоритма, компонентами которого являются входящие в него ПП, взаимодействующие друг с другом путем передачи сообщений через каналы.

Пусть задан РП $\mathcal{P}$. Будем использовать следующие обозначения и предположения:

- $Var(\mathcal{P}) = \bigcup\limits_{P \in \mathcal{P}} Var(P)$, множества $X(\mathcal{P})$, $\bar{X}(\mathcal{P})$, $\tilde{X}(\mathcal{P})$, $\hat{X}(\mathcal{P})$ определяются аналогично,
- будем предполагать, что

$$\text{компоненты семейства } \{\bar{X}(P) \cup \hat{X}(P) \mid P \in \mathcal{P}\} \text{ дизъюнктны и не пересекаются с } \tilde{X}(\mathcal{P}) \tag{6}$$

  (если это не так, то заменим каждый из компонентов $P$ семейства $\mathcal{P}$ на равный ему в том смысле, который указан в конце пункта 1.2.7, так, чтобы свойство (6) выполнялось),

- РП $\mathcal{P}$ может обозначаться записью
  - $\{P_1, \ldots, P_n\}$, если $I = \{1, \ldots, n\}$ (в случае $n = 1$ скобки могут быть опущены, то есть вместо $\{P_1\}$ пишется $P_1$), или
  - $P^*$, если $I$ — множество натуральных чисел, и все ПП, входящие в $\mathcal{P}$, совпадают с $P$,

- запись $\mathcal{P}_\dagger$ обозначает РП $\{\mathcal{P}, P_\dagger\}$,

- если $\{\mathcal{P}_i \mid i \in I\}$ — семейство РП, и для любого $i \in I$ РП $\mathcal{P}_i$ является семейством ПП вида $\{P_{i'} \mid i' \in I_i\}$, где множества индексов $I_i$ $(i \in I)$ дизъюнктны (если это не так, то заменим их на соответствующие дизъюнктные копии), то запись $\{\mathcal{P}_i \mid i \in I\}$ обозначает также РП $\{P_{i'} \mid i' \in \bigsqcup_{i \in I} I_i\}$.

### 1.3.2. Понятие состояния распределенного процесса

Пусть задан РП $\mathcal{P}$.

С о с т о я н и е м РП $\mathcal{P}$ называется семейство $s = \{s_P \mid P \in \mathcal{P}\}$ состояний ПП, входящих в $\mathcal{P}$, такое, что для любого $c \in \mathit{Channels}$ все множества в семействе $\{[c]_{s_P} \mid P \in \mathcal{P}\}$ одинаковы (обозначаем их $[c]_s$).

Пусть $s = \{s_P \mid P \in \mathcal{P}\}$ — состояние РП $\mathcal{P}$. Тогда

- $s$ называется н а ч а л ь н ы м состоянием РП $\mathcal{P}$, и обозначается $0_\mathcal{P}$, если $s_P = 0_P$ для любого $P \in \mathcal{P}$,

- $at_s = \{at_{s_P} \mid P \in \mathcal{P}\}$, $[\mathcal{P}]_s = \bigcup_{P \in \mathcal{P}} [P]_s$, $\langle \mathcal{P} \rangle_s = Tm([\mathcal{P}]_s)$,

- $\theta_s$ обозначает связывание из $\Theta([\mathcal{P}]_s)$ такое, что

$$\forall\, P \in \mathcal{P},\ \forall\, x \in [P]_s \quad \theta_{s_P}(x) = \theta_s(x),$$

существование такого связывания следует из (6).

Понятия значения выражения и значения формулы в состоянии РП определяются аналогично соответствующим понятиям для ПП.

Для всяких $\varphi, \psi \in Fm$ запись $\varphi \leqslant \psi$ означает, что для каждого РП $\mathcal{P}$ и каждого состояния $s$ РП $\mathcal{P}$ верна импликация $s \models \varphi \Rightarrow s \models \psi$.

Если формулы $\varphi, \psi \in Fm$ таковы, что $\varphi \leqslant \psi$ и $\psi \leqslant \varphi$, то будем рассматривать такие формулы как одинаковые. Если формулы $\varphi$ и $\psi$ одинаковы, то будем обозначать этот факт записью $\varphi = \psi$.

### 1.3.3. Выполнение распределенного процесса

Пусть задан РП $\mathcal{P}$. Выполнение РП $\mathcal{P}$ представляет собой недетерминированное чередование выполнений ПП, входящих в $\mathcal{P}$. На каждом шаге выполнения РП $\mathcal{P}$

- только один ПП из $\mathcal{P}$ выполняет активный переход, и
- остальные ПП из $\mathcal{P}$ выполняют пассивные переходы.

Выполнение РП $\mathcal{P}$ можно определить как порождение последовательности состояний этого РП (начиная с начального состояния $0_{\mathcal{P}}$), в которой каждое состояние $s$, не являющееся последним в этой последовательности, связано со следующим состоянием $s'$ отношением перехода, что означает следующее: существует $P \in \mathcal{P}$ такое, что

$$s_P \xrightarrow{P} s'_P, \quad \forall\, P' \in \mathcal{P} \setminus \{P\} \;\; s_{P'} \to s'_{P'}, \\ \text{где } s = \{s_P \mid P \in \mathcal{P}\},\ s' = \{s'_P \mid P \in \mathcal{P}\}. \quad (7)$$

Свойство (7) обозначается записью $s \xrightarrow{\alpha_P} s'$, где $\alpha = \alpha_{s'_P}$.

Множество всех состояний РП $\mathcal{P}$ можно рассматривать как граф, в котором существует ребро из $s$ в $s'$ с меткой $\alpha_P$ тогда и только тогда когда $s \xrightarrow{\alpha_P} s'$. Обозначение ПП $P$ в метке $\alpha_P$ можно опускать.

Для каждой пары состояний $s, s'$ РП $\mathcal{P}$ запись $s \to s'$ означает, что $s$ связано с $s'$ отношением перехода, и запись $s \Rightarrow s'$ означает, что существует последовательность $s_1, \ldots, s_n$ состояний такая, что $s_1 = s$, $s_n = s'$, и $s_i \to s_{i+1}$ для всех $i = 1, \ldots, n-1$.

Состояние $s$ РП $\mathcal{P}$ называется достижимым, если $0_{\mathcal{P}} \Rightarrow s$. Множество достижимых состояний РП $\mathcal{P}$ обозначается записью $\Sigma_{\mathcal{P}}$.

Если задан путь $\pi$ из $0_{\mathcal{P}}$ в $s$, и $s'$ — какое-либо состояние, входящее в $\pi$, то мы будем обозначать этот факт записью $s' \leqslant_\pi s$. Запись $s' <_\pi s$ обозначает, что $s' \leqslant_\pi s$ и $s' \neq s$. Если путь $\pi$ ясен из контекста, то обозначение этого пути в записях $\leqslant_\pi$ и $<_\pi$ может быть опущено.

## 1.4. Теорема для доказательства свойства соответствия

Теорема, излагаемая в этом параграфе, может использоваться для доказательства свойства соответствия протоколов

аутентификации, которое имеет следующий смысл: если один из участников протокола аутентификации после выполнения этого протокола пришел к выводу, что другой участник этого протокола является подлинным (то есть объявленные им свое имя и параметры совпадают с его реальными именем и параметрами), то это действительно так. Доказываемая ниже теорема применяется для обоснования того, что если РП $\mathcal{P}$ использует для взаимодействия только открытый канал $\circ$, и в некотором состоянии $s \in \Sigma_{\mathcal{P}}$ в этом канале содержится сообщение, содержащее подтерм вида $k(e)$, где ключ $k$ недоступен в этом состоянии для некоторого ПП $P$, входящего в $\mathcal{P}$, то в некотором состоянии $s' <_\pi s$ другой ПП $P' \neq P$ из $\mathcal{P}$ послал в открытый канал $\circ$ сообщение, содержащее тот же самый подтерм $k(e)$.

Теорема 1. *Пусть заданы РП $\mathcal{P}$, такой, что $Var(\mathcal{P})_{\mathbf{C}} = \{\circ\}$, ПП $P \in \mathcal{P}$, множество $E \subseteq \langle\mathcal{P}\rangle_0$, не содержащее открытых ключей, и состояние $s \in \Sigma_{\mathcal{P}}$, причем $s \models E \perp_{\mathbf{K}} P$, и $[\circ]_s$ содержит терм с подтермом $k(e)$, где $k \in E_{\mathbf{K}}$.*

*Тогда для каждого пути $\pi$ из начального состояния 0 РП $\mathcal{P}$ в состояние $s$ существует ПП $P' \in \mathcal{P} \setminus \{P\}$ такой, что $\pi$ содержит ребро вида*

$$\dot{s} \xrightarrow{(!\dot{e})_{P'}} s', \quad \text{где } k(e) \subseteq \dot{e}^{\dot{s}}. \tag{8}$$

### 1.5. Схемы распределенных процессов

Пусть задан РП $\mathcal{P}$. Зависимости между действиями в $\mathcal{P}$ можно выразить в виде схемы РП $\mathcal{P}$, в которой каждый ПП $P \in \mathcal{P}$ представляется нитью, то есть вертикальной линией, на которой выделены точки, соответствующие вершинам из $P$. Пример схемы РП представлен диаграммой (10). В целях большей наглядности будем указывать в схемах РП горизонтальную черту над любым обозначением какой-либо переменной $x$, если она рассматривается как элемент множества $\bar{X}(P)$, то есть эта переменная обозначается $\bar{x}$.

## 2. Верификация протокола Yahalom

В этом параграфе рассматривается пример КП, который можно верифицировать на основе предлагаемого подхода путем использо-

вания теоремы 1.

### 2.1. Описание протокола Yahalom

КП Yahalom предназначен для аутентификации (то есть проверки подлинности) агентов, взаимодействующих по открытому каналу $\circ$, и передачи сеансовых ключей между этими агентами.

Предполагается что заданы множество агентов $Ag$, а также агент $J$, называемый доверенным посредником, данные агенты могут взаимодействовать друг с другом по открытому каналу $\circ$. Каждый агент $A \in Ag$ имеет общий секретный ключ $k_{AJ}$ с доверенным посредником $J$, на котором $A$ и $J$ могут шифровать и дешифровать сообщения, используя симметричную систему шифрования, причем только $A$ и $J$ знают ключ $k_{AJ}$.

В каждом сеансе КП Yahalom принимают участие следующие агенты: инициатор $A \in Ag$, доверенный посредник $J$, и респондер $B \in Ag$. Каждый агент из $Ag$ в одних сеансах может быть инициатором, а в других — респондером. Один и тот же агент может быть и инициатором, и респондером в одном и том же сеансе (то есть возможно, что $A = B$). Выполнение сеанса КП Yahalom с инициатором $A$, респондером $B$ и доверенным посредником $J$ представляет собой совокупность четырех пересылок сообщений:

$$\begin{array}{lllll} 1) & A \to B & : & A, n_A \\ 2) & B \to J & : & B, k_{BJ}(A, n_A, n_B) \\ 3) & J \to A & : & k_{AJ}(B, k, n_A, n_B), k_{BJ}(A, k) \\ 4) & A \to B & : & k_{BJ}(A, k), k(n_B) \end{array} \tag{9}$$

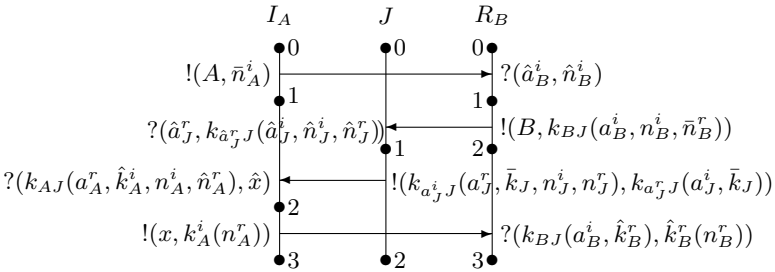
Пересылки в (9) имеют следующий смысл:

1) $A$ посылает $B$ запрос на аутентификацию и генерацию сеансового ключа $k$, запрос состоит из имени $A$ и нонса $n_A$,
2) $B$ посылает $J$ запрос на генерацию сеансового ключа $k$, в запрос он включает свое имя, имя агента $A$, для связи с которым нужен этот ключ, полученный нонс $n_A$, и свой нонс $n_B$,
3) $J$ генерирует сеансовый ключ $k$ и посылает $A$ пару сообщений, из первого сообщения $A$ может извлечь сеансовый ключ $k$, а второе предназначено для того, чтобы $A$ переслал его $B$,

4) $A$ посылает $B$ пару сообщений,

- первое из которых было получено им от $J$, агент $B$ может извлечь из этого сообщения сеансовый ключ $k$, и
- используя ключ $k$, агент $B$ дешифрует второе сообщение, если результат дешифрования совпадает с его нонсом $n_B$, то это является для него доказательством того, что отправителем этого сообщения был именно $A$.

Формальное описание сеанса КП Yahalom изображается схемой



(10)

В этой схеме левая и правая нити соответствуют ПП $I_A$ и $R_B$, описывающим поведение инициатора $A$ и респондера $B$ соответственно, средняя нить соответствует ПП, описывающему поведение посредника $J$, этот ПП обозначается тем же символом $J$. Смысл переменных в этих ПП усматривается из сопоставления действий в этих ПП с соответствующими действиями в (9). Верхний индекс $i$ или $r$ при какой-либо переменной означает, что она предположительно содержит информацию об инициаторе $(i)$ или респондере $(r)$ данного сеанса.

Считаем, что $Agent(I_A) = A$, $Agent(R_B) = B$, $Agent(J) = J$.

РП $\mathcal{P}$, соответствующий КП Yahalom, имеет вид

$$\mathcal{P} = \{\{I^*_A \mid A \in Ag\}, \{R^*_B \mid B \in Ag\}, J^*\}. \tag{11}$$

Ниже мы будем использовать следующее обозначение запись $s \models E \perp_{\mathbf{K}} e$ обозначает утверждение $\forall\, x \in E_{\mathbf{X}}\ \ x \perp_{\mathbf{K},E} e^s$.

### 2.2. Свойства протокола Yahalom

На основе предлагаемого подхода: могут быть верифицированы следующие свойства РП (11):

- секретность ключей и нонсов $n_B^r$:

$$\forall\, s \in \Sigma_{\mathcal{P}_\dagger} \quad s \models E \perp_{\mathbf{K}} P_\dagger, \quad \text{где } E = \{k_{BJ}, k_J, n_B^r \mid B \in Ag\} \quad (12)$$

- аутентификация инициатора перед респондером: для любых $R_B \in \mathcal{P}$, $s \in \Sigma_{\mathcal{P}_\dagger}$, если $s \models at_{R_B} = 3$, то существует $I_A \in \mathcal{P}$ такое, что

$$s \models \{at_{I_A} = 3, a_A^r = B, a_B^i = A, n_A^i = n_B^i, n_A^r = n_B^r, k_A^i = k_B^r\}, \quad (13)$$

- аутентификация респондера перед инициатором: для любых $I_A \in \mathcal{P}$, $s \in \Sigma_{\mathcal{P}_\dagger}$, если $s \models at_{I_A} = 2$, то существует $R_B \in \mathcal{P}$ такое, что

$$s \models \{at_{R_B} = 2, a_A^r = B, a_B^i = A, n_A^i = n_B^i, n_A^r = n_B^r\}. \quad (14)$$

### Заключение

В настоящей работе была построена новая модель КП, и показан пример ее использования для решения задач верификации свойств целостности, секретности и соответствия.

Для дальнейшей деятельности по развитию данной модели и основанных на ней методов верификации можно назвать следующие задачи:

- развитие языков спецификаций свойств КП, позволяющих выражать например свойства нулевого разглашения в КП аутентификации, свойства неотслеживаемости в КП электронных платежей, свойства анонимности и правильности подсчета голосов в КП электронного голосования, и разработка методов верификации свойств, выражаемых на этих языках,
- построение методов автоматизированного синтеза КП по описанию свойств, которым они должны удовлетворять.

### Список литературы

[1] *Cortier, V.* Formal Models and Techniques for Analyzing Security Protocols: A Tutorial / V. Cortier, S. Kremer // Foundations and Trends in Programming Languages. — 2014. — Vol. 1, №3. — P. 151–267.

[2] *Lowe, G.* An attack on the Needham-Schroeder public key authentication protocol // Information Processing Letters. — 1995. — Vol. 56, №3. — P. 131–133.

[3] *Needham, R.* Using encryption for authentication in large networks of computers / R. Needham, M. Schroeder // Communications of the ACM. — 1978. — Vol. 21, №12. — P. 993–999.

### Библиографическая ссылка

*Миронов, А. М.* Верификация криптографических протоколов // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 213–234.
https://doi.org/10.26456/mfcsics-21-31

### Сведения об авторах

Андрей Михайлович Миронов
Университет Иннополис;
Лидирующий Исследовательский Центр. Доцент

*Россия, 420500, Иннополис, ул. Университетская, 1*
*E-mail: amironov66@gmail.com*

УДК 519.688, 519.682.3, 004.42
AMS MSC2020: 68W40, 68Q25, 97N80

# Автоматическое тестирование студенческих программ

**Новиков М. Д.**

Московский государственный университет им. М. В. Ломоносова, факультет вычислительной математики и кибернетики

Аннотация. В статье описываются две системы, предназначенные для тестирования программ, выполняемых студентами первого курса факультета ВМК МГУ в рамках практикума на ЭВМ. Системы разрабатываются на факультете ВМК с 2017 года. Первая система тестирует программы на языке Паскаль, а вторая — на языке Ассемблера.

Ключевые слова: язык Паскаль, язык Ассемблера, тестирующие программы, программирование, информатика.

## Введение

Языки программирования Паскаль и Ассемблер изучаются студентами факультета ВМК МГУ на первом курсе в течение многих лет; задачи для практических занятий по Паскалю берутся, в основном, из книг [9] и [8], а задачи по Ассемблеру берутся из книг [1] и [2]. В этих учебных пособиях собрано много задач на различные темы; они охватывают все основные конструкции языков Паскаль и Ассемблер. Возникает проблема тестирования написанных студентами программ, то есть проверки правильности их работы на различных наборах исходных данных. Проверка простых программ обычно не вызывает затруднений, а проверка сложных программ предполагает ввод с клавиатуры нетривиальных исходных данных и сверка результатов, выдаваемых студенческой программой, с правильными результатами. Это занимает много времени и часто не гарантирует выявление всех ошибок. Указанные сложности тестирования программ побудили автора к созданию систем, позволяющих

производить тестирование в автоматическом режиме, то есть запускать тестируемую программу для различных, заранее заготовленных тестовых данных и сравнивать результаты, выдаваемые программой, с эталонными (правильными) результатами.

## 1. Требования к студенческим программам

В формулировках задач из [1, 2, 8, 9] указаны требования, предъявляемые к той или иной студенческой программе. К некоторым задачам надо написать полную программу на Паскале или Ассемблере; алгоритмы для решения других задач должны быть оформлены в виде фрагмента программы, процедуры или функции. Для некоторых задач из [1, 2, 8, 9] дополнительно указаны требования, предъявляемые к методам их решения: например, необходимо описать подпрограмму с определенным набором параметров, реализовать рекурсивный алгоритм, использовать конкретный тип данных языка Паскаль или Ассемблера и т. д. Целью таких требований является обучение студентов не только составлять программы, но и осваивать различные приемы программирования и конструкции изучаемых языков программирования.

## 2. Описание систем автоматического тестирования программ

Возможности обеих систем следующие.

1) Можно тестировать либо полную программу, либо фрагмент программы. Если (по условию задачи) необходимо написать именно фрагмент программы, то система тестирования добавляет к ней блок, дополняющий ее до полной программы, которая затем тестируется. В системе тестирования программ на Паскале есть также возможность тестировать полную программу наряду с автономным тестированием входящей в нее процедуры или функции. Это необходимо, если по условию задачи требуется составить полную программу и описать внутри нее подпрограмму заданного вида.
2) Проверяется выполнение требований к алгоритмам, указанным в [1, 2, 8, 9]:

а) запрет или, наоборот, обязательное использование каких-либо конструкций языка Паскаль или Ассемблера;

б) реализация заданного конкретного алгоритма решения задачи.

Для проверки выполнения требований к алгоритмам производится анализ тестируемой программы. При этом подсчитывается количество вхождений определенных слов в программу, например, while, if, procedure и др. По результатам анализа делается заключение о соответствии программы указанным требованиям.

3) Проверяется корректность выполнения тестируемых программ:

а) в задачах на использование внешних файлов (Паскаль) проверяется закрытие их в программе по окончании работы с ними; такая ошибка обычно не фиксируется операционной системой;

б) в задачах на динамические структуры данных (Паскаль) проверяется корректность использования динамической памяти; для этого подсчитывается количество обращений к процедурам New и Dispose в процессе выполнения программы;

в) в задачах на составление процедур и функций (Паскаль) проверяется отсутствие в теле подпрограммы глобальных переменных и способ вызова фактических параметров — по значению или по ссылке;

г) в задачах на составление программ обработки символьных данных (Ассемблер) проверяется установка флага направления просмотра строк.

Для проверки корректности выполнения тестируемых программ к ним добавляются специальные операторы: например, чтобы проверить закрытие файла в процедуре, система тестирования вставляет операторы close(f); I:=ioresult;if I=0 then ... (файл не был закрыт) после обращения к этой процедуре.

4) Реализованы также некоторые дополнительные проверки тестируемых программ, в частности:

а) в задачах, допускающих различные формы вывода результата, любой допустимый вывод считается правильным;

б) проверяется количество используемых циклов и досрочные выходы из циклов (отсутствие избыточных вычислений) там, где это необходимо по условию задачи;

в) в задачах на использование ограниченного типа данных (Паскаль) в качестве тестов предлагаются числа, выходящие за допустимый диапазон значений. Возникающая при этом ошибка периода выполнения считается верным ответом.

Каждая тестируемая программа выполняется на заданных наборах исходных данных. По окончании тестирования выдается результат — исходные данные, выданные программой ответы, правильные ответы, общее количество правильных ответов и количество ошибок с указанием типа каждой ошибки.

Обе системы автоматического тестирования написаны в среде Delphi и могут выполняться в ОС Windows; для трансляции тестируемых программ используются компиляторы Free-Pascal и Masm 6.14. Системы могут быть установлены на локальном компьютере и не требуют доступа в Интернет. К настоящему времени составлены тесты примерно к 550 задачам по Паскалю и к 300 задачам по Ассемблеру. Планируется составить тесты к задачам из [1, 2, 8, 9], еще не вошедшим в системы тестирования и реализовать тестирование полной программы на языке Ассемблера наряду с автономным тестированием входящей в нее процедуры.

Системы автоматического тестирования были представлены на двух конференциях: на конференции, посвященной памяти Н. П. Трифонова [5] и на конференции Ломоносовские чтения [6]. По теме опубликованы три статьи [3, 4, 7].

## Заключение

Использование систем показало их эффективность. Раньше для проверки студенческой программы преподавателю приходилось вручную вводить сложные исходные данные и сверять результат с пра-

вильным; при этом часто обнаруживались ошибки. Теперь же студенты отлаживают свои программы самостоятельно, принося на проверку уже готовые варианты.

**Список литературы**

[1] *Бордаченкова, Е. А.* Задания практикума. 1 курс / Е. А. Бордаченкова, А. А. Панферов. — М. : МАКС Пресс, 2016. — 48 с.

[2] *Бордаченкова, Е. А.* Задачи и упражнения по языку Ассемблера MASM. — М. : МАКС Пресс, 2020. — 92 с.

[3] *Новиков, М. Д.* Система автоматического тестирования программ, написанных на языке Паскаль // Альманах современной науки и образования. — 2017. — №6. — С. 68–71.

[4] *Новиков, М. Д.* Автоматизированный практикум по языку программирования Паскаль // Наука России: цели и задачи. Сборник научных трудов по материалам XVII международной научно-практической конференции 10 октября 2019 г. Часть 1 Изд. НИЦ «Л-Журнал», 2019. — С. 26–31.

[5] *Новиков, М. Д.* Автоматическое тестирование программ на языке Паскаль // Программирование и вычислительная математика. Тезисы докладов конференции памяти Н. П. Трифонова / под ред. С. А. Абрамова, А. В. Столярова. — М. : МАКС Пресс, 2020. — С. 34–36.

[6] *Новиков, М. Д.* Система автоматического тестирования студенческих программ на языке Ассемблера // Ломоносовские чтения. Тезисы докладов. Секция вычислительной математики и кибернетики. — М. : Изд-во Московского ун-та, 2021. — С. 116–117.

[7] *Новиков, М. Д.* Тестирование программ на языке Паскаль, использующих динамическую память // Eastern-European Scientific Journal. — 2021. — Т. 1, №3 (67). — С. 45–47.

[8] *Пильщиков, В. Н.* Задания практикума на ЭВМ. 1 курс / В. Н. Пильщиков, Н. П. Трифонов. — М. : Издательский отдел факультета ВМК МГУ, 2001. — 34 с.

[9] *Пильщиков, В. Н.* Язык Паскаль: упражнения и задачи. — М. : Научный мир, 2003. — 224 с.

**Библиографическая ссылка**

*Новиков, М. Д.* Автоматическое тестирование студенческих программ // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 235–240.
https://doi.org/10.26456/mfcsics-21-32

**Сведения об авторах**

Михаил Дмитриевич Новиков
Московский государственный университет им. М. В. Ломоносова, факультет вычислительной математики и кибернетики. Старший научный сотрудник

*Россия, 119991 ГСП-1 Москва, Ленинские горы, МГУ им. М. В. Ломоносова, д.1, стр. 52, ф-т ВМК*
*E-mail: novikov_57@mail.ru*

УДК 510.64
AMS MSC2020: 03B60, 03B42

# Топологические модели логик HC и H4[1]

**Оноприенко А. А.**

Тверской государственный университет

Аннотация. Рассматривается пропозициональный фрагмент HC совместной логики задач и высказываний QHC, введенной С. А. Мелиховым. Предлагаются топологические модели этой логики. Также рассмотрены топологические модели логики H4, являющейся расширением так называемой «lax logic». Для всех этих моделей доказана теорема о корректности и полноте.

Ключевые слова: неклассические логики, топологическая семантика.

## Введение

В математике еще со времен Евклида было принято разделять утверждения на две группы: высказывания (теоремы, предположения) и задачи (геометрические построения, нахождение корней уравнения). Теоремы необходимо доказывать, а в задачах находить общий метод, приводящий к решению в любом частном случае. Таким образом, в математической науке имеются два вида знания: «знать, что нечто истинно» (то есть иметь знание об истинности некоторых утверждений) и «знать, как что-либо делать» (то есть знать общий метод решения задач).

А. Н. Колмогоров рассматривал интерпретацию интуиционистской логики высказываний как логики задач [1]. А. Н. Колмогоров критически исследовал интуиционистскую логику и указывал, что ее объекты — это, по существу, задачи, а не теоретические высказывания, и поэтому интуиционистская логика должна быть заменена исчислением задач. По замыслу А. Н. Колмогорова работа [1] должна

[1]Работа выполнена при финансовой поддержке РНФ, проект 21-18-00195. Автор является стипендиатом Фонда развития теоретической физики и математики «БАЗИС».

была стать предпосылкой к созданию «единого логического аппарата», работающего одновременно с объектами двух типов: задачами и высказываниями.

С. А. Мелихов ввел в рассмотрение объединенную логику задач и высказываний QHC [3,4]. В этой логике каждый предикатный символ и каждая формула имеет один из двух сортов: высказывание либо задача. Формулы сорта высказывание (задача) подчиняются всем законам классического (интуиционистского) исчисления предикатов. Формулы разных сортов связаны друг с другом при помощи двух модальностей: ? и ! . Применив модальность ! к высказыванию $p$, мы получим задачу $!p$ — «доказать высказывание $p$». С другой стороны, применив модальность ? к задаче $\alpha$, мы получим высказывание $?\alpha$ — «задача $\alpha$ имеет решение». Эти модальности связаны между собой следующими аксиомами и правилами вывода:

1) $!(p \to q) \to (!p \to !q)$;
2) $?(\alpha \to \beta) \to (?\alpha \to ?\beta)$;
3) $\dfrac{p}{!p}$;
4) $\dfrac{\alpha}{?\alpha}$;
5) $?!p \to p$;
6) $\alpha \to !?\alpha$;
7) $\neg!0$.

## 1. Топологические модели HC

Определим модели логики HC (пропозиционального фрагмента логики QHC) следующим образом.

- Зафиксируем топологическое пространство $X$ и его всюду плотное подмножество $A$.
- Классическая часть интерпретируется на булевой алгебре подмножеств множества $A$.
- Интуиционистская часть интерпретируется на топологическом пространстве $X$ стандартным образом [5].

- Интерпретация модальностей задается следующим образом:
  $|?\alpha| = A \cap |\alpha|$; $|!p| = X \setminus \mathrm{Cl}(A \setminus |p|)$.

Для данной семантики получены следующие результаты.

Теорема 1 (корректность). *Если замкнутая формула выводима в* HC, *то она истинна в любой топологической модели логики* HC.

Теорема 2 (полнота). *Если формула $\varphi$ логики* HC *истинна в любой топологической модели логики* HC, *то $\varphi$ выводима в* HC.

Замечание 1. *Приведенные модели являются обобщением моделей логики* QHC, *которые ввел С. А. Мелихов под названием «модели Эйлера – Тарского» [4].*

Замечание 2. *Если положить $A = X$ и рассматривать только классическую часть логики* HC *с производной модальностью* $\Box = ?!$, *то получатся топологические модели логики* S4.

## 2. Логика H4

Логика H4 — расширение интуиционистской пропозициональной логики модальностью $\nabla$, для которой выполнены следующие аксиомы:

1) $\alpha \to \nabla\alpha$;
2) $\nabla\nabla\alpha \to \nabla\alpha$;
3) $\nabla\bot \to \bot$;
4) $\nabla(\alpha \to \beta) \to (\nabla\alpha \to \nabla\beta)$.

Определим топологические модели логики H4 следующим образом.

- Зафиксируем топологическое пространство $X$ и его всюду плотное подмножество $A$.
- Интепретация интуиционистских связок и кванторов стандартна [5].
- Интерпретация модальности задается следующим образом:
  $|\nabla\varphi| = X \setminus \mathrm{Cl}(A \setminus |\varphi|)$.

Для данной семантики получены аналогичные результаты.

Теорема 3 (корректность). *Если замкнутая формула выводима в* H4*, то она истинна в любой топологической модели логики* H4.

Теорема 4 (полнота). *Если формула $\varphi$ логики* H4 *истинна в любой топологической модели логики* H4*, то $\varphi$ выводима в* H4.

Замечание 3. *Если рассматривать только интуиционистскую часть* HC *с производной модальностью $\nabla = !?$, то из топологических моделей логики* HC *получатся топологические модели логики* H4.

## Заключение

В настоящей работе рассмотрена топологическая семантика совместной логики задач и высказываний HC, являющаяся обобщением топологических семантик классической логики, интуиционистской логики, логики S4, а также логики H4. Обозначим некоторые открытые вопросы в данной области исследования.

1) Выполнена ли теорема о полноте для логик HC и H4, если ограничиться топологическим пространством $\mathbb{R}$ и его всюду плотным подмножеством $\mathbb{Q}$?
2) Логика H4 является расширением так называемой «lax logic» [2] дополнительной аксиомой $\nabla\bot \to \bot$. Будет ли выполнена теорема о полноте для «lax logic», если исключить в определении топологической семантики требование о всюду плотности множества $A$?
3) Выполнена ли теорема о полноте для логик QHC и QH4 — предикатных вариантов логик HC и H4?

## Список литературы

[1] *Колмогоров, А. Н.* Избранные труды. Математика и механика. — М. : Наука, 1985. — 470 с.

[2] *Fairtlough, M.* Quantified lax logic / M. Fairtlough, M. Walton. — Sheffield, 1997. — 78 p. — (Tech. Rep. / Department of Computer Science, University of Sheffield; CS-97-11.)

[3] *Melikhov, S. A.* A Galois connection between classical and intuitionistic logics. I: Syntax. — URL: arXiv:1312.2575. — Загл. с титул. экрана.

[4] *Melikhov, S. A.* A Galois connection between classical and intuitionistic logics. II: Semantics. — URL: arXiv:1504.03379. — Загл. с титул. экрана.

[5] *Tarski, A.* Der Aussagenkalkül und die Topologie // Fundamenta Mathematicae. — 1938. — Vol. 31. — P. 103–134.

### Библиографическая ссылка

*Оноприенко, А. А.* Топологические модели логик HC и H4 // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 241–245.
https://doi.org/10.26456/mfcsics-21-33

### Сведения об авторах

Оноприенко Анастасия Александровна
Тверской государственный университет.

*Россия, 170002, Тверь, Садовый пер., д. 35*
*E-mail: ansidiana@yandex.ru*

УДК 510.643, 510.649
AMS MSC2020: 03B20, 03B25, 03B45

# Неразрешимость логик с унарным предикатом и двумя переменными[1]

**Рыбаков М. Н.**[*], **Шкатов Д. П.**[**]

[*]Тверской государственный университет
[**]Тверской государственный университет;
University of the Witwatersrand, Johannesburg

Аннотация. Обсуждается вопрос об алгоритмической сложности неклассических предикатных логик в языке с одной унарной предикатной буквой и двумя переменными. Показано, что если в логике нет формул, ограничивающих высоту (иногда и ширину) ее шкал Крипке, то, в зависимости от остальных ограничений, проблема принадлежности формул логике в таком языке будет неразрешима, неперечислима или даже неарифметична.

Ключевые слова: неклассические логики, логики первого порядка, разрешимость, рекурсивная перечислимость.

## Введение

Неклассические логики предикатов обычно содержат классическую логику первого порядка **QCl** как естественный фрагмент,[2] и следовательно, неразрешимы; более того, для их неразрешимости часто достаточно двух одноместных предикатных букв [5] или же двух предметных переменных [4]. Уменьшение числа предметных переменных до одной, как правило, приводит к разрешимости [2, 6], а вот уменьшение числа предикатных букв до одной — не обязательно [1, 3]. Авторам данной работы удалось установить, что во многих случаях при «соединении» этих двух условий (одна унарная

[1]Работа выполнена в Тверском государственном университете при финансовой поддержке Российского научного фонда, проект 21-18-00195.

[2]Иногда не содержат в явном виде, но **QCl** погружается в них, как, например, происходит в случае суперинтуиционистских логик.

буква и две переменные) получается $\Sigma^0_1$-, $\Pi^0_1$- или даже $\Pi^1_1$-трудный фрагмент исходной логики [8–11], и цель данной работы — показать ключевую идею, лежащую в основе соответствующих доказательств.

## 1. Основные определения

Пусть интуиционистский предикатный язык содержит счетное множество предметных переменных, счетное множество предикатных букв любой валентности, связки $\wedge$, $\vee$, $\to$, $\bot$, а также кванторные символы $\forall$ и $\exists$, а модальный предикатный язык содержит еще и модальность $\Box$. Определим $\neg$ и $\Diamond$ как обычные сокращения: $\neg\varphi = \varphi \to \bot$, $\Diamond\varphi = \neg\Box\neg\varphi$.

Шкалой Крипке называем пару $\mathfrak{F} = (W, R)$, где $W$ — непустое множество миров и $R$ — бинарное отношением достижимости на $W$. Предикатным шкалой называем набор $\mathfrak{F}_D = (\mathfrak{F}, D)$, где $\mathfrak{F}$ — шкала Крипке, $D$ — семейство предметных областей миров из $W$, содержащее для каждого $w \in W$ непустое множество $D_w$, причем $D_w \subseteq D_u$ для любого $u \in R(w)$; в случае когда $D_w = D_u$ для любых $w \in W$ и $u \in R(w)$, говорят, что $\mathfrak{F}_D$ — шкала с постоянными областями.

Модальная модель Крипке — это набор $\mathfrak{M} = (\mathfrak{F}_D, I)$, где $\mathfrak{F}_D$ — предикатная шкала, а $I$ — интерпретация предикатных букв в мирах из $W$, то есть функция, которая $n$-арной букве $P$ и миру $w \in W$ ставит в соответствие $n$-арный предикат $I(P, w) = P^{I,w}$ в $D_w$. Заметим, что если мы определим $I_w$, положив $I_w(P) = I(P, w)$, то пара $(D_w, I_w)$ будет классической моделью. Истинность $\models$ в мирах такой модели определяется стандартно, в частности,

$$\mathfrak{M}, w \models \Box\varphi(\bar{a}) \quad \leftrightharpoons \quad \begin{array}{l}\mathfrak{M}, w' \models \varphi(\bar{a}) \\ \text{для любого } w' \in R(w).\end{array}$$

Интуиционистская модель Крипке — это набор $\mathfrak{M} = (\mathfrak{F}_D, I)$, где $\mathfrak{F}_D$ — предикатная шкала, отношение достижимости в которой рефлексивно, транзитивно и антисимметрично, а $I$ — интерпретация предикатных букв в мирах из $W$, для которой выполнено условие наследственности: если $u \in R(w)$ и $P^{I,w}(\bar{a})$, то $P^{I,u}(\bar{a})$. Истинность $\Vdash$ в мирах такой модели также определяется стандартно, в частности,

$$\mathfrak{M}, w \Vdash \forall x\, \varphi(x, \bar{a}) \quad \leftrightharpoons \quad \begin{array}{l}\mathfrak{M}, w' \Vdash \varphi(b, \bar{a}) \\ \text{для любых } w' \in R(w) \text{ и } b \in D_{w'}.\end{array}$$

Истинность формулы в модели, предикатной шкале, шкале Крипке, классе шкал определяется стандартно. Шкалу называем конечной, если множество ее миров конечно; $L_{wfin}$ — логика класса конечных шкал логики $L$.

## 2. Модальные логики классов шкал неограниченной ширины и высоты

Если шкалы логики не ограничены по ширине,[3] то формулу вида $Q(x, y)$ можно промоделировать формулой $\Diamond(P_1(x) \wedge P_2(y))$; эта идея изложена в [5] и известна как трюк Крипке. За счет такого моделирования мы оставляем в формулах лишь унарные буквы: уже в классической логике все $n$-арные буквы моделируются одной бинарной, а бинарная моделируется в модальных логиках двумя унарными. Трюк Крипке работает не только в шкалах бесконечной ширины: важно, чтобы из текущего мира было достижимо достаточно много других миров.

Пусть $\varphi$ — формула, содержащая унарные буквы $P_1, \ldots, P_n$. Чтобы промоделировать формулы вида $P_1(x), \ldots, P_n(x)$, зафиксируем унарную букву $P$. Пусть $\mathfrak{F}_n = (W_n, R_n)$, где $W_n = \{0, \ldots, n\}$ и $R_n = \{(k, k+1) : k \in W_n \setminus \{n\}\}$.

Пусть $\mathfrak{M} = (\mathfrak{F}_D, I)$ — модель Крипке и $w$ — мир в ней. Для каждого элемента $a \in D_w$ определены значения $P_1^{I,w}(a), \ldots, P_n^{I,w}(a)$; возьмем в качестве предметных областей миров шкалы $\mathfrak{F}_n$ множество $D_w$, и определим модель $\mathfrak{M}_w$ на шкале $\mathfrak{F}_n$ с этими областями в соответствии с эквивалентностью

$$\mathfrak{M}_w, k \models P(a) \iff \mathfrak{M}, w \models P_k(a).$$

В результате получим, что $\mathfrak{M}_w$ моделирует мир $w$ модели $\mathfrak{M}$.

Проделаем это для каждого мира $w$ исходной модели $\mathfrak{M}$, беря каждый раз в описанной конструкции не саму шкалу $\mathfrak{F}_n$, а ее новую копию. Добавим получившиеся таким способом модели к исходной модели (объединим множества миров и отношения достижимости), а также для каждого мира $w$ модели $\mathfrak{M}$ сделаем достижимым корень модели $\mathfrak{M}_w$ из $w$.

[3] Высота тоже имеет значение, но фактически часто бывает достаточно шкал высоты два.

Теперь заметим, что корень модели $\mathfrak{M}_w$ описывается формулой $\alpha = \Diamond^n\Box\bot$, а миры исходной модели $\mathfrak{M}$ — формулой $\Diamond\alpha$. И тогда, чтобы сказать, что в $w$ истинно $P_k(a)$, достаточно сказать, что из корня модели $\mathfrak{M}_w$ за $k$ шагов достижим мир, где истинно $P(a)$; таким образом, $P_k(x)$ можно промоделировать формулой $\Diamond(\alpha \wedge \Diamond^k P(x))$.

Некоторые технические детали добавляются, когда отношение достижимости должно обладать дополнительными свойствами, например, рефлексивностью, симметричностью, транзитивностью. Первые два особых сложностей не представляют, а вот в транзитивном случае удается провести подобное моделирование лишь для моделей с условием наследственности[4] (чего оказывается достаточно для доказательства неразрешимости соответствующих логик). Наличие сразу всех трех указанных свойств приводит к ситуации, где похожее моделирование в мономодальном языке проделать не удалось (но можно в бимодальном).

В итоге получаем следующую теорему. Пусть *bf* — формула Баркан (семантически *bf* обеспечивает постоянство областей).

Теорема 1. *Пусть $L$ — логика, содержащая* **QK** *и содержащаяся в одной из логик* **QGL**, **QGrz**, **QKTB**. *Тогда логики $L$ и $L \oplus bf$ являются $\Sigma^0_1$-трудными в языке с одной унарной предикатной буквой и двумя предметными переменными, а логики $L_{wfin}$ и $L_{wfin} \oplus bf$ являются $\Pi^0_1$-трудными в языке с одной унарной предикатной буквой и тремя предметными переменными.*

Замечание 1. *В доказательстве этой теоремы оказывается не нужной полнота по Крипке, поэтому логики в указанных интервалах могут быть любыми; можно даже брать произвольные множества формул, удовлетворяющие указанному условию.*

## 3. Модальные логики классов шкал ограниченной ширины

Мы будем говорить здесь лишь о логиках линейных шкал (отношение достижимости при этом предполагается транзитивным); в случае, когда ширина шкал больше единицы, приведенная ниже аргументация также пройдет. Ситуация с логиками классов конечных шкал при этом особо не отличается от той, когда линейность не

[4] То есть $u \in R(w)$ и $P^{I,w}(\bar{a})$ влечет $P^{I,u}(\bar{a})$. Альтернативно можно взять условие $u \in R(w)$ и $P^{I,u}(\bar{a})$ влечет $P^{I,w}(\bar{a})$.

требовалась, и мы не будем ее рассматривать. В линейной шкале с бесконечной возрастающей цепью миров тоже проходит описанный выше трюк Крипке. Таким образом, мы снова можем отталкиваться от языка, содержащего лишь унарные предикатные буквы.

В этом случае моделирование формул $P_1(x), \ldots, P_n(x)$ возможно, например, благодаря следующей несложной идее. Пусть, для простоты, у нас имеется шкала $(\mathbb{N}, <)$. Выделим миры вида $m \cdot (n+1)$ и будем смотреть на них как на миры исходной модели на шкале $(\mathbb{N}, <)$, между любыми двумя соседними из которых «вставлены» еще $n$ миров. Чтобы выделить эти миры, сделаем в мире $m \cdot (n+1)$ истинной фиксированную пропозициональную переменную $q$, а также $P(a_m)$, предварительно (до моделирования бинарных букв унарными) описав бинарное отношение $\lhd$, в соответствии с которым и выбраны элементы $a_i$ для $i \in \mathbb{N}$: $a_i \lhd a_{i+1}$. Теперь, чтобы промоделировать $P_k(a)$, в мире $m \cdot (n+1)$ достаточно выполнить условие типа $q \wedge P(a_m) \wedge \Diamond(P(a) \wedge \Diamond^k(q \wedge P(a_{m+1})) \wedge \neg\Diamond^{k+1}(q \wedge P(a_{m+1})))$.

Здесь мы видим три элемента ($a$, $a_m$ и $a_{m+1}$), но, тем не менее, можно обойтись формулами с двумя переменными. Дальнейшая детализация всей конструкции[5] требует некоторой технической работы, здесь мы излагаем лишь ключевую идею, лежащую в основе результатов, изложенных в [9].

Теорема 2. *Пусть $L$ — логика шкалы $(\mathbb{N}, R)$, где $R$ — бинарное отношение, лежащее между $<$ и $\leqslant$. Тогда логики $L$ и $L \oplus bf$ являются $\Pi^1_1$-трудными в языке с одной одноместной предикатной буквой, одной пропозициональной переменной и двумя предметными переменными.*

Замечание 2. *Этот результат можно распространить и на другие порядки, например, заменив $\mathbb{N}$ на $\mathbb{Q}$ или $\mathbb{R}$, но тогда вместо $\Pi^1_1$-трудности мы можем гарантировать лишь $\Sigma^0_1$-трудность при аналогичных ограничениях на средства языка. При переходе к темпоральным логикам типа* **QLTL**, **QCTL**, **QCTL**$^*$ *для обснования $\Pi^1_1$-трудности пропозициональная переменная не требуется.*

[5] Нужно описать свойства отношения $\lhd$, связать их с отношением достижимости в шкале, использовать для моделирования неразрешимой проблемы.

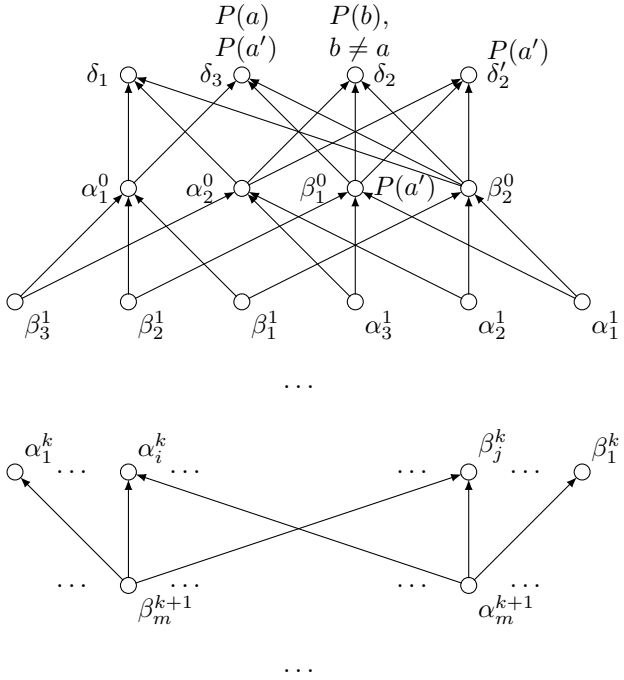



Рис. 1. Модель $\mathfrak{N}_a^w$

## 4. Суперинтуиционистские логики

Интуиционистская логика **QInt** неразрешима в языке с двумя предметными переменными [4], но нам важно, что в исходном доказательстве можно промоделировать константу $\bot$ позитивной формулой $\forall x\, Q(x)$. Это позволяет применить некоторую модификацию трюка Крипке, оставаясь в позитивном фрагменте **QInt**. Далее используем модификацию «пропозициональной» конструкции из [7], суть которой в следующем. Чтобы промоделировать в мире $w$ интуиционистской модели $\mathfrak{M}$ формулы $P_1(x), \ldots, P_n(x)$, для каждого $a \in D_w$ построим модель $\mathfrak{N}_a^w = (W_0, R_0, D^w, I_a)$, как на рис. 1, где $a'$ — некоторый фиксированный элемент, отличный от $a$, и $D_u^w = D_w$ для каждого $u \in W_0$.

Определим формулы, в некотором смысле[6] описывающие верхние миры этой модели:

$$\begin{array}{lcl}
D_1 & = & \exists x\, P(x);\\
D_2(x) & = & \exists x\, P(x) \to P(x);\\
D_3(x) & = & P(x) \to \forall x\, P(x);\\
A_1^0(x) & = & D_2(x) \to D_1 \vee D_3(x);\\
A_2^0(x) & = & D_3(x) \to D_1 \vee D_2(x);\\
B_1^0(x) & = & D_1 \to D_2(x) \vee D_3(x);\\
B_2^0(x) & = & A_1^0(x) \wedge A_2^0(x) \wedge B_1^0(x) \to D_1 \vee D_2(x) \vee D_3(x);\\
A_1^1(x) & = & A_1^0(x) \wedge A_2^0(x) \to B_1^0(x) \vee B_2^0(x);\\
A_2^1(x) & = & A_1^0(x) \wedge B_1^0(x) \to A_2^0(x) \vee B_2^0(x);\\
A_3^1(x) & = & A_1^0(x) \wedge B_2^0(x) \to A_2^0(x) \vee B_1^0(x);\\
B_1^1(x) & = & A_2^0(x) \wedge B_1^0(x) \to A_1^0(x) \vee B_2^0(x);\\
B_2^1(x) & = & A_2^0(x) \wedge B_2^0(x) \to A_1^0(x) \vee B_1^0(x);\\
B_3^1(x) & = & B_1^0(x) \wedge B_2^0(x) \to A_1^0(x) \vee A_2^0(x).
\end{array}$$

Формулы, описывающие остальные миры, определим рекурсивно. Миры $\alpha_i^k$ и $\beta_i^k$ называем мирами уровня $k$; число миров уровня $k$ равно $2n_k$, где $n_0 = 2$, $n_1 = 3$, $n_{k+1} = (n_k - 1)^2$. Пусть формулы, описывающие миры уровня $k$, где $k \geqslant 1$, определены. Пусть $i$, $j$ и $m$ таковы, что из $\alpha_m^{k+1}$ и $\beta_m^{k+1}$ достижимы $\alpha_i^k$ и $\beta_j^k$; тогда положим

$$\begin{array}{lcl}
A_m^{k+1}(x) & = & A_1^k(x) \to B_1^k(x) \vee A_i^k(x) \vee B_j^k(x);\\
B_m^{k+1}(x) & = & B_1^k(x) \to A_1^k(x) \vee A_i^k(x) \vee B_j^k(x).
\end{array}$$

Ключевым наблюдением здесь является следующее.

Лемма 3. *Для любого мира и модели $\mathfrak{N}_a^w$ верны следующие эквивалентности:*

$$\begin{array}{lcl}
\mathfrak{N}_a, u \not\Vdash A_m^k(a) & \iff & uR_0\alpha_m^k;\\
\mathfrak{N}_a, u \not\Vdash B_m^k(a) & \iff & uR_0\beta_m^k.
\end{array}$$

Это наблюдение позволяет промоделировать $P_k(x)$ формулой $A_k^{n+1}(x) \vee B_k^{n+1}(x)$ и повторить конструкцию, похожую на описанную для модального случая, при этом модель $\mathfrak{N}_a^w$ нужно брать для каждого набора $(w, a, k)$, когда $\mathfrak{M}, w \not\models P_k(a)$ или $k = n+1$. Отметим также, что фактически нужна не вся модель $\mathfrak{N}_a^w$, а лишь ее верхняя часть вплоть до уровня $n + 1$, которая является конечной.

[6] См. лемму 3.

В итоге можно доказать следующую теорему. Пусть $cd$ — формула, требующая постоянства областей.

Теорема 4. *Пусть $L$ — логика, лежащая между* **QInt** *и* **QKC**. *Тогда логики $L$ и $L + cd$ являются $\Sigma_1^0$-трудными в языке с одной унарной предикатной буквой и двумя предметными переменными, а $L_{wfin}$ и $L_{wfin} + cd$ являются $\Pi_1^0$-трудными в языке с одной унарной предикатной буквой и тремя предметными переменными.*

**Список литературы**


[1] *Маслов, Ю. Г.* Неразрешимость в конструктивном исчислении предикатов некоторых классов формул, содержащих только одноместные предикатные переменные / С. Ю. Маслов, Г. Е. Минц, В. П. Оревков // Доклады АН СССР. — 1965. — Т. 163, № 2. — С. 295–297.

[2] *Минц, Г. Е.* О некоторых исчислениях модальной логики // Логические и логико-математические исчисления. Тр. МИАН СССР. — 1968. — Т. 98. — С. 88–111.

[3] *Gabbay, D.* Semantical Investigations in Heyting's Intuitionistic Logic. — Dordrecht : Springer, 1981. — 287 p.

[4] *Kontchakov, R.* Undecidability of first-order intuitionistic and modal logics with two variables / R. Kontchakov, A. Kurucz, M. Zakharyaschev // Bulletin of Symbolic Logic. — 2005. — Vol. 11, №3. — P. 428–438.

[5] *Kripke, S.* The undecidability of monadic modal quantification theory // Zeitschrift für Matematische Logik und Grundlagen der Mathematik. — 1962. — №8. — P. 113–116.

[6] *Ono, H.* On some intuitionistic modal logics // Publications of the Research Institute for Mathematical Sciences. — 1977. — Vol. 13, №3. — С. 687–722.

[7] *Rybakov, M.* Complexity of intuitionistic propositional logic and its fragments // Journal of Applied Non-Classical Logics. — 2008. — Vol. 18, №2-3. — P. 267–292.

[8] *Rybakov, M.* Undecidability of first-order modal and intuitionistic logics with two variables and one monadic predicate letter /

M. Rybakov, D. Shkatov // Studia Logica. — 2019. — Vol. 107, №4. — P. 695–717.

[9] *Rybakov, M.* Algorithmic properties of first-order modal logics of the natural number line in restricted languages / M. Rybakov, D. Shkatov // N. Olivetti, R. Verbrugge, S. Negri, G. Sandu, editors. Advances in Modal Logic. — 2020. — Vol. 13. — P. 523–539.

[10] *Rybakov, M.* Algorithmic properties of first-order modal logics of finite Kripke frames in restricted languages / M. Rybakov, D. Shkatov // Journal of Logic and Computation. — 2020. — Vol. 30, №7. — P. 1305–1329.

[11] *Rybakov, M.* Algorithmic properties of first-order superintuitionistic logics of finite Kripke frames in restricted languages / M. Rybakov, D. Shkatov // Journal of Logic and Computation. — 2021. — Vol. 31, №2. — P. 494–522.

### Библиографическая ссылка

*Рыбаков, М. Н.* Неразрешимость логик с унарным предикатом и двумя переменными / М. Н. Рыбаков, Д. П. Шкатов // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 246–254.
https://doi.org/10.26456/mfcsics-21-34

### Сведения об авторах

1. Михаил Николаевич Рыбаков
Тверской государственный университет. Доцент

*Россия, 170100, г. Тверь, ул. Желябова, 33*
*E-mail: m_rybakov@mail.ru*

2. Дмитрий Петрович Шкатов
Тверской государственный университет;
University of the Witwatersrand, Johannesburg. Senior Lecturer

*South Africa, Johannesburg, WITS2050, Private Bag 3*
*E-mail: shkatov@gmail.com*

УДК 510.624
AMS MSC2020: 03B70

# Элиминация оператора частичной фиксированной точки[1]

**Секорин В. С.**

Тверской государственный университет

Аннотация. В работе рассмотрена семантика частичной фиксированной точки для бесконечных алгебраических систем. Для нее показано, что элиминировать оператор частичной фиксированной точки можно только в тех теориях, в которых любой такой оператор зацикливается за конечное число шагов.

Ключевые слова: частичная фиксированная точка, алгебраическая система, семантика.

## Введение

Разнообразные методы математической логики находят все более широкое применение при решении задач проектирования и анализа программного обеспечения. Одним из разделов математической логики, который наиболее тесным образом связан с информатикой, является теория логических языков. Например, логические языки используются в системах управления базами данных. В них они применяются в качестве средства извлечения информации из базы данных. Но стоит отметить, что многие простые, но имеющие большое практическое значение [1] свойства являются невыразимыми в логике первого порядка.

Эта причина обосновывает тот факт, что логика первого порядка и различные ее расширения постоянно изучаются. Среди одних из самых распространенных расширений можно выделить оператор фиксированной точки. Существует несколько видов таких операторов: инфляционной фиксированной точки, наименьшей фиксированной точки и частичной фиксированной точки. Самым

[1]Работа выполнена при финансовой поддержке РФФИ, проект 20-01-00435.

общим из этих операторов является оператор частичной фиксированной точки (PFP-оператор). Отметим, что предложен этот оператор был Ю. Гуревичем в работе [2]. В книге [3] Л. Либкина содержится подробное изложение свойств PFP-оператора для конечных алгебраических систем.

Необходимо отметить, что операции базы данных могут выполняться не только над элементами самой базы данных, но и над произвольными элементами универсума. Это тоже может увеличить выразительные возможности языка первого порядка, но незначительно [4]. При совместном использовании этих двух возможностей возникает ситуация, при которой применение PFP-оператора происходит для бесконечных алгебраических систем. В данной работе мы рассмотрим семантику оператора частичной фиксированной точки, которая заключается в том, что формула считается истинной, когда набор принадлежит предикату на всех шагах, начиная с некоторого [5].

Возникает вопрос: в каких случаях можно элиминировать PFP-оператор. В качестве основного результата доказано утверждение о том, что в теории можно элиминировать PFP-оператор тогда и только тогда, когда любой PFP-оператор в этой теории зацикливается.

## 1. Основные определения

Определение 1 (Формула логики частичной фиксированной точки, [3], [5]). *Будем называть формулой* PFP-*логики формулу, которая построена по правилам логики первого порядка с использование оператора частичной фиксированной точки* PFP*: если $\varphi(\bar{x}, \bar{y})$ — формула со свободными переменными $\bar{x}$ и $\bar{y}$, содержащая несигнатурный предикатный символ $Q$, то $\mathrm{PFP}_{Q(\bar{y})}(\varphi)$ — формула исходной сигнатуры, содержащая свободные переменные $\bar{x}$ и $\bar{y}$. При этом длина $\bar{y}$ совпадает с местностью $Q$.*

*Пусть $\mathfrak{A}$ — это алгебраическая система. Зафиксируем значения переменных $\bar{x}$ как $\bar{d} \in |\mathfrak{A}|$. Построим последовательность множеств $Q_i^{\bar{d}}$ следующим образом. Пусть*

$$Q_0^{\bar{d}} = \varnothing; \quad Q_{i+1}^{\bar{d}} = \{\bar{y} \in |\mathfrak{A}| \mid (\mathfrak{A}, Q_i^{\bar{d}}) \models \varphi(\bar{d}, \bar{y})\},$$

*для $i \in \omega$.*

*Значением частичной фиксированной точки является следующее множество $Q^{\bar{d}}_{\forall}$. Множеству $Q^{\bar{d}}_{\forall}$ принадлежат только те $\bar{y}$, для которых существует $i$ такой, что $\bar{y} \in Q_j$ для всех $j > i$. Следовательно, для этих $\bar{y}$ формула $\mathrm{PFP}^{\forall}_{Q(\bar{y})}(\varphi)(\bar{d}, \bar{y})$ будет истинной.*

*Если существуют такие натуральные и неравные $i, j$, что выполнено $Q^{\bar{d}}_i = Q^{\bar{d}}_j$, то будем говорить, что* $\mathrm{PFP}^{\forall}$*-оператор зацикливается или сходится.*

Пример 1. *Рассмотрим алгебраическую систему, носителем которой является множество целых чисел, и одноместный функциональный символ $s^{(1)}$ обозначает число на единицу большее. Тогда следующая формула будет истинна тогда и только тогда, когда $v \leqslant w$ $\mathrm{PFP}^{\forall}_{Q(x)}(\theta)(v, w)$, где*

$$\theta(v, x) \equiv x = v \vee Q(x) \vee (\exists y)\big(Q(y) \wedge x = s(y)\big).$$

*На первом шаге в предикат $Q$ попадет только число $v$, то есть $Q_1 = \{v\}$. На всех последующих шагах будет выполнено:*

$$Q_{i+1} = \{x \mid Q_i(x) \text{ или } (\exists y)\big(Q_i(y) \wedge x = s(y)\big)\},$$

*таким образом на $i+1$-ом шаге в предикат попадут те и только те числа, которые были на предыдущем шаге или которые на единицу больше чисел предыдущего шага. То есть на $i+1$ шаге будет выполнено: $Q_{i+1} = \{v, \ldots, v+i\}$. Таким образом, данный* $\mathrm{PFP}^{\forall}$*-оператор не зацикливается, не сходится, так как на каждом шаге мы добавляем новое число.*

Определение 2 (Элиминация PFP-оператора). *Теория $T$ допускает элиминацию $PFP$-оператора, если для любой формулы $\varphi$, содержащей* PFP*-операторы, существует эквивалентная ей в $T$ формула $\psi$ без* PFP*-операторов.*

## 2. Основные результаты

Основной результат, который мы докажем в этой части, заключается в том, что элиминация $\mathrm{PFP}^{\forall}$-оператора возможна в теории $T$ тогда и только тогда, когда в этой теории любой $\mathrm{PFP}^{\forall}$-оператор сходится.

Теорема 1. *Теория $T$ допускает элиминацию* $\mathrm{PFP}^{\forall}$*-оператора тогда и только тогда, когда любой* $\mathrm{PFP}^{\forall}$*-оператор зацикливается в* $T$.

Доказательство. Докажем теорему в прямую сторону. Допустим, что $\mathrm{PFP}^{\forall}$-оператор не сходится за конечное число шагов, то есть существует бесконечное количество попарно неравных множеств $Q_0, Q_1, Q_2, \ldots$

Пусть

$$\varphi'(\bar{y}) \equiv (\varphi)^{Q(\bar{t})}_{(\exists \bar{u},\bar{v})P(\bar{t},\bar{u},\bar{v})}(\bar{y}).$$

Построим $\mathrm{PFP}_P(\eta)(\bar{x},\bar{y},\bar{z})$, где

$$\eta(\bar{x},\bar{y},\bar{z}) \equiv \varphi'(\bar{x}) \wedge \\ \wedge \big[(\forall \bar{s},\bar{t})\big(\varphi'(\bar{s}) \wedge \neg\varphi'(\bar{t}) \vee \neg\varphi'(\bar{s}) \wedge \varphi'(\bar{t}) \to \bar{y} = \bar{s} \wedge \bar{z} = \bar{t}\big) \vee \\ \vee (\exists \bar{s}) P(\bar{s},\bar{y},\bar{t})\big].$$

Множество $P_i$ будет содержать тройку $(\bar{x},\bar{y},\bar{z})$ тогда и только тогда, когда для некоторого натурального $j \leqslant i$ выполнено $Q_j(\bar{y}) \neq Q_j(\bar{z})$, то есть когда на этом или на одном из предыдущих шагов построения по $Q$ один из этих наборов принадлежал множеству, а второй нет. Первый аргумент предиката $P$ будет использован для сохранения значения $Q$. Для краткости при помощи $R$ обозначим следующие множества: $R_0 = \overline{\varnothing}$, $R_i = (\exists \bar{x})P_{i-1}(\bar{x},\bar{y},\bar{z})$. Таким образом, множеству $R_i$ будут принадлежать такие пары $(\bar{u},\bar{v})$, что на некотором шаге $j < i$ построения по предикату $Q$ было выполнено $Q_j(\bar{u}) \neq Q_j(\bar{v})$.

Покажем, что $R_i$ неограниченно возрастает. Допустим, что это не выполняется, то есть существует некоторое натуральное $j_0$ такое, что для всех $j \geqslant j_0$ выполняется $R_j = R_{j+1}$. Это возможно в том и только том случае, когда на $j-1$ шаге построения PFP по $Q$ не нашлось новой пары $\bar{u},\bar{v}$ такой, что $Q_{j-1}(\bar{u}) \neq Q_{j-1}(\bar{v})$. Каждый $i$-ый шаг построения по $Q$ при $i < j_0$ разбивает носитель на две части: элементы, принадлежащие $Q_i$, и элементы, не принадлежащие $Q_i$. Таким образом, мы получим не более $2^{j_0}$ частей: $A_0, \ldots, A_{2^{j_0}-1}$. На всех последующих шагах $j > j_0$ множество $Q_j$ будут являться объединением некоторых таких частей $A$. Всего таких объединений не больше чем $2^{2^j}$. Следовательно, на шаге $j_0 + 2^{2^j} + 1$ будет получено множество $Q$, совпадающее с одним из предыдущих. Из этого следует,

что и дальше они будут повторяться. Получили противоречие с тем, что существует бесконечно много различных множеств $Q_i$.

Тогда $R$ задает дискретный предпорядок на $(\bar{x}, \bar{y})$: будем считать $(\bar{x}, \bar{y}) < (\bar{u}, \bar{v})$, если существует такой $i$, что $(\bar{x}, \bar{y}) \in R_i$ и $(\bar{u}, \bar{v}) \notin R_i$. Покажем, как такой предпорядок можно определить при помощи $\mathrm{PFP}^{\forall}$-оператора. Пусть

$$\eta'(\bar{a}, \bar{b}, \bar{c}) \equiv \big(\eta\big)^{P(\bar{r},\bar{s},\bar{t})}_{(\exists\bar{x},\bar{y},\bar{u},\bar{v})S(\bar{r},\bar{s},\bar{t},\bar{x},\bar{y},\bar{u},\bar{v})}(\bar{a}, \bar{b}, \bar{c}).$$

Тогда $(\bar{x}, \bar{y}) < (\bar{u}, \bar{v}) \Leftrightarrow (\exists\bar{r}, \bar{s}, \bar{t})\big(\mathrm{PFP}^{\forall}_S(\xi)(\bar{r}, \bar{s}, \bar{t}, \bar{x}, \bar{y}, \bar{u}, \bar{v})\big)$, где

$$\begin{aligned}\xi(\bar{r}, \bar{s}, \bar{t}, \bar{x}, \bar{y}, \bar{u}, \bar{v}) \equiv \eta'(\bar{r}, \bar{s}, \bar{t}) \wedge \\ \wedge \big[(\forall\bar{a}, \bar{b}, \bar{c}, \bar{d})\big((\exists\bar{g})\eta'(\bar{g}, \bar{c}, \bar{d}) \wedge \neg(\exists\bar{e}, \bar{f}, \bar{g}, \bar{h}, \bar{k})S(\bar{e}, \bar{c}, \bar{d}, \bar{f}, \bar{g}, \bar{h}, \bar{k}) \wedge \\ \wedge (\exists\bar{e}, \bar{f}, \bar{g}, \bar{h}, \bar{k})S(\bar{e}, \bar{a}, \bar{b}, \bar{f}, \bar{g}, \bar{h}, \bar{k}) \to \bar{x} = \bar{a} \wedge \bar{y} = \bar{b} \wedge \bar{u} = \bar{c} \wedge \bar{v} = \bar{d}\big) \vee \\ \vee (\exists\bar{a}, \bar{b}, \bar{c})S(\bar{a}, \bar{b}, \bar{c}, \bar{x}, \bar{y}, \bar{u}, \bar{v})\big]\end{aligned}$$

Используя $R_i$, определим формулу $\psi(\bar{x}, \bar{y})$ истинную тогда и только тогда, когда не выполнено $R_i(\bar{x}, \bar{y})$ для всех $R_i$. Допустим, что формула $\psi$ эквивалентна некоторой формуле $\theta$, которая не содержит PFP-оператора. Рассмотрим множество формул

$$B = \{\neg\theta(\bar{x}, \bar{y})\} \cup \{\neg R_i(\bar{x}, \bar{y}) \mid \text{для всех } i\}.$$

Покажем, что это множество конечно совместно. Так как в любом конечном подмножестве найдется формула $\neg R_i(\bar{x}, \bar{y})$ с максимальным $i$. При помощи $j$ обозначим максимальный из этих $i$. Тогда найдется такая пара $(\bar{u}, \bar{v})$, что она будет принадлежать множеству $R_{j+1}$, но не будет принадлежать множествам $R_i$ с меньшими номерами. Следовательно, любое конечное подмножество $B$ будет иметь модель, тогда само множество является конечно совместным. Таким образом, по теореме компактности мы получаем, что множество формул $B$ имеет модель. В этом случае выполнена формула $\neg\theta(\bar{x}, \bar{y})$, говорящая о том, что хотя бы одна из формул вида $R_i(\bar{x}, \bar{y})$ выполнена, и одновременно выполнены формулы $\neg R_i(\bar{x}, \bar{y})$ для всех $i$, то есть ни одна из формул вида $R_i(\bar{x}, \bar{y})$ не выполнена. Противоречие.

Теперь докажем теорему в обратную сторону. Любой $\mathrm{PFP}^{\forall}$-оператор зацикливается в $T$, рассмотрим произвольный из них.

Пусть $n$ — это максимальное число шагов, за которое зацикливается выбранный $\mathrm{PFP}^{\forall}$-оператор на любом наборе аргументов. Заметим, что количества шагов, необходимые для зацикливания на некотором наборе, не могут неограниченно возрастать, так как в этом случае по теореме компактности нашелся бы набор, для которого количество шагов было бы бесконечным. Таким образом, мы получили, что произвольно выбранный $\mathrm{PFP}^{\forall}$-оператор не сходится, что противоречит тому, что любой такой оператор зацикливается. Следовательно, существует конечно много попарно неравных $Q_i : Q_0, \ldots, Q_n$. Тогда $Q_{n+1} = Q_k$ для некоторого натурального $k \leqslant n$. Покажем, что любой зацикливающийся $\mathrm{PFP}^{\forall}$-оператор, можно элиминировать формулой $\chi(\bar{x}) \equiv \bigvee\limits_{k=0}^{n} \big((\forall \bar{y})(Q_k(\bar{y}) \leftrightarrow Q_{n+1}(\bar{y})) \wedge \bigwedge\limits_{i=k}^{n+1} Q_i(\bar{x})\big)$. Для этого докажем, что для произвольного сходящегося $\mathrm{PFP}^{\forall}$-оператора выполнено $\mathrm{PFP}^{\forall}_{Q}(\varphi)(\bar{a}) \Leftrightarrow \chi(\bar{a})$. Если выполнено $\mathrm{PFP}^{\forall}_{Q}(\varphi)(\bar{a})$, то $\bar{a} \in Q_i$ для всех $i$ начиная с некоторого. Следовательно, $\bar{a} \in Q_i$ для всех $i$ входящих в цикл, начинающегося с некоторого шага $k \leqslant n$, тогда получаем, что выполнено $\bigwedge\limits_{i=k}^{n+1} Q_i(\bar{a})$. Получаем, что выполнено $\chi(\bar{a})$. Если выполнено $\chi(\bar{a})$, то есть $\bar{a}$ принадлежит всем $Q$ в некотором цикле. Следовательно, не существует такого натурального $l > n$, что $\bar{a} \notin Q_l$. Тогда получаем, что по определению $\mathrm{PFP}^{\forall}$-оператора выполнено $\mathrm{PFP}^{\forall}_{Q}(\varphi)(\bar{a})$. Таким образом, мы показали, что если $\mathrm{PFP}^{\forall}$-оператор зацикливается в $T$ за $n$ шагов, то можно его элиминировать в теории $T$. □

**Заключение**

Мы продемонстрировали, что $\mathrm{PFP}^{\forall}$-оператор можно элиминировать в некоторой теории $T$ тогда и только тогда, когда в этой теории $T$ любой $\mathrm{PFP}^{\forall}$-оператор зацикливается. В качестве вопросов, которые могут представлять интерес для изучения в будущих работах по данной тематике, можно выделить поиск других свойств теорий при которых $\mathrm{PFP}^{\forall}$-оператор будет возможно или невозможно элиминировать.

**Список литературы**

[1] *Aho, A. V.* Universality of data retrieval languages / A. Aho, J. D. Ullman // Proc. of 6th Symp. on Principles of Programming Languages. — New York, N. Y. : ACM Press, 1979. — P. 110–120.

[2] *Gurevich, Y.* Fixed-point extensions of first-order logic / Y. Gurevich, S. Shelah // Annals of Pure and Applied Logic. — Vol. 32. — 1986. — P. 265–280.

[3] *Libkin, L.* Elements of Finite Model Theory. — Berlin : Springer, 2004. — 314 p.

[4] *Дудаков, С. М.* Трансляционные результаты для языков запросов в теории баз данных / С. М. Дудаков, М. А. Тайцлин // Успехи математических наук. — 2006. — Т. 61, вып. 2 (368). — С. 3–66.

[5] *Секорин, В. С.* Об эквивалентности двух семантик PFP-оператора // Вестник ТвГУ. Серия: Прикладная математика. — 2020. — № 3. — С. 41–49.

**Библиографическая ссылка**

*Секорин, В. С.* Элиминация оператора частичной фиксированной точки // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 255–261.
https://doi.org/10.26456/mfcsics-21-35

**Сведения об авторах**

Всеслав Станиславович Секорин
Тверской государственный университет. Аспирант

*Россия, 170100, г. Тверь, Тверская обл., ул. Желябова, 33*
*E-mail: vssekorin@gmail.com*

УДК 512.644
AMS MSC2020: 15A06

# О сводимости систем линейных уравнений

**Селиверстов А. В.**

Институт проблем передачи информации им. А. А. Харкевича РАН

Аннотация. Рассмотрена задача распознавания, существует ли $(0,1)$-решение для системы линейных уравнений с целыми коэффициентами. Алгебраический подход позволяет уточнить структуру множества трудных входов. С другой стороны, количество $(0,1)$-решений одного линейного уравнения от $n$ переменных равно количеству $(0,1)$-решений системы линейных уравнений, коэффициенты которых органичены многочленом от суммы размеров двоичных записей коэффициентов исходного уравнения.

Ключевые слова: линейное уравнение, двоичное решение, вычислительная сложность.

Для многих задач, хотя известные алгоритмы имеют высокую вычислительную сложность в худшем случае, существуют так называемые генерические алгоритмы, работающие без ошибок и быстро принимающие или отвергающие почти любой вход, но уведомляющие об отказе от решения на малой доле входов [2, 3].

Рассмотрим задачу распознавания: даны $m \times n$ матрица $A$ и вектор $\mathbf{b}$ с целыми коэффициентами, узнать, существует ли $(0,1)$-решение у системы линейных уравнений $A\mathbf{x} = \mathbf{b}$.

В случае, когда все элементы $m \times n$ матрицы $A$ и вектора $\mathbf{b}$ неотрицательные, метод динамического программирования позволяет перечислить все $(0,1)$-решения системы неравенств $A\mathbf{x} \leqslant \mathbf{b}$. Вычислительная сложность линейно зависит от общего числа таких решений. При $m > c \log_2 n$ для некоторой константы $c$ и некоторых предположениях о распределении коэффициентов, среднее число решений полиномиально ограничено, следовательно, все решения легко найти. Доказательство, которое предложил Н. Н. Кузюрин [1], основано на оценке хвостов биномиального распределения. В этом частном случае легко выбрать те $(0,1)$-решения, на которых неравенства обращаются в равенства.

С другой стороны, задача распознавания $(0,1)$-решения у системы $A\mathbf{x} = \mathbf{b}$ для любых $A$ и $\mathbf{b}$ может быть сведена к ее частному случаю, когда система состоит всего из одного линейного уравнения [4]. Он известен как задача о разбиении множества. Тогда $(0,1)$-решение одного линейного уравнения может быть найдено за псевдополиномиальное время. Однако в общем случае уравнение имеет большие коэффициенты, следовательно, этот подход не дает эффективного генерического алгоритма.

Случай, когда $2m \leqslant n \leqslant m \log_2 n$ и уравнения линейно независимые, остается вычислительно трудным в худшем случае даже при малых абсолютных величинах коэффициентов всех уравнений системы. Алгебраический подход позволяет уточнить структуру множества трудных входов. Согласно [3], когда число переменных $n$ и число уравнений $m$ удовлетворяют неравенству вида $m > n - \sqrt{2n - o(n)}$, набор трудных входов включается в множество нулей многочлена от коэффициентов уравнений. Этот многочлен отличен от константы и определяется числами $n$ и $m$. Новый результат уточняет оценку.

Теорема 1. *Существуют сублинейная функция $s = o(n)$ и генерический алгоритм полиномиального времени, который для всех положительных целых чисел $n$ и $m$, удовлетворяющих неравенству $m > n - \sqrt{6n - s(n)}$, и для почти каждого набора $m$ линейных форм $\ell_j(x_0, \cdots, x_{n-m})$, где $j > n - m$, допускает лишь такой вход, для которого не существует $(0,1)$-решения системы уравнений $x_j = \ell_j(1, x_1, \cdots, x_{n-m})$. Более того, для указанных $n$ и $m$ этот алгоритм не отвергает вход и существует такой отличный от константы многочлен степени $O(\sqrt{n^3})$ от коэффициентов линейных форм $\ell_j$, что, если алгоритм дает уведомление об отказе, то этот многочлен обращается в нуль.*

Хотя все NP-полные задачи сводимы по Карпу друг к другу, образ такой сводимости может составлять малую долю всех случаев. Поэтому полиномиальная в среднем разрешимость некоторой NP-полной задачи не влечет существование такого алгоритма для других задач из класса NP. Рассмотрим пример такой сводимости.

Лемма 2. *Существует такая константа $c > 0$, что за полиномиальное время для данного числа $s \geqslant 2$ можно найти список различных простых чисел $p_k$, произведение которых превосходит число $s$, где каждое число удовлетворяет неравенству $p_k < c \log_2 s$.*

Доказательство. Алгоритм реализует решето Эратосфена. Обозначим через $\vartheta(x)$ функцию Чебышева, равную натуральному логарифму произведения простых чисел, которые не превосходят $x$. Согласно [5], для $x \geqslant 2$ выполнено неравенство

$$|\vartheta(x) - x| \leqslant \frac{151.3x}{\ln^4 x}.$$

Так получается верхняя оценка для простых чисел $p_k$. □

Теорема 3. *Дано линейное уравнение от $n$ переменных над $\mathbb{Z}$. За полиномиальное время вычислимы матрица $A$ и вектор $\mathbf{b}$ над $\mathbb{Z}$, для которых исходное уравнение и система уравнений $A\mathbf{x} = \mathbf{b}$ имеют одинаковое количество $(0,1)$-решений. Более того, все коэффициенты новой системы неотрицательные и ограничены сверху многочленом от $n$ и общего размера двоичных записей коэффициентов.*

Доказательство. Обозначим через $a_1x_1 + \cdots + a_nx_n = a_0$ исходное уравнение, где все коэффициенты $a_0, \ldots, a_n$ — ненулевые целые числа. Обозначим через $s$ сумму абсолютных величин коэффициентов $s = |a_0| + |a_1| + \cdots |a_n|$. По лемме 2 за полиномиальное время вычисляется список различных простых чисел $p_1 < \cdots < p_r$, удовлетворяющий неравенству $s < p_1 \cdot p_2 \cdots p_{r-1} \cdot p_r$.

По китайской теореме об остатках, исходное уравнение имеет те же $(0,1)$-решения, что и система сравнений от $n$ переменных

$$\begin{cases} a_1x_1 + \cdots a_nx_n & \equiv & a_0 & \pmod{p_1} \\ \cdots & \cdots & \cdots & \cdots \\ a_1x_1 + \cdots a_nx_n & \equiv & a_0 & \pmod{p_r} \end{cases}$$

Обозначим через $a_j \bmod p_k$ остаток от деления числа $a_j$ на $p_k$, принимающий значения от нуля до $p_k - 1$. В свою очередь, каждое из сравнений $a_1x_1 + \cdots a_nx_n \equiv a_0 \pmod{p_k}$ имеет столько же $(0,1)$-решений, что и следующее уравнение над $\mathbb{Z}$, зависящее от новых переменных $y_{k0}, \ldots, y_{ku}$, где $u = \lfloor \log_2 n \rfloor - 1$,

$$\sum_{j=1}^{n} (a_j \bmod p_k) x_j - p_k \sum_{\ell=0}^{u} 2^\ell y_{k\ell} = a_0 \bmod p_k$$

Так получается система из $r$ уравнений над $\mathbb{Z}$, но к прежним $n$ переменным добавилось еще $r\lfloor \log_2 n \rfloor$ новых переменных. Замена переменных $y_{k\ell} = 1 - x_{k\ell}$, меняет знаки у коэффициентов. □

Пример 1. *Рассмотрим уравнение* $x_1 + x_2 - x_3 = 2$. *Сумма абсолютных величин коэффициентов равна* $s = 5$. *Поэтому достаточно взять два простых числа* $p_1 = 2$ *и* $p_2 = 3$. *Система сравнений*

$$\begin{cases} x_1 + x_2 + x_3 \equiv 0 \pmod 2 \\ x_1 + x_2 + 2x_3 \equiv 2 \pmod 3 \end{cases}$$

*сводится к системе уравнений над* $\mathbb{Z}$ *с двумя новыми переменными* $y_1$ *и* $y_2$, *где каждое уравнение имеет тот же набор* $(0,1)$*-решений, что и соответствующее сравнение*

$$\begin{cases} x_1 + x_2 + x_3 - 2y_1 = 0 \\ x_1 + x_2 + 2x_3 - 3y_2 = 2 \end{cases}$$

*Замена* $y_1 = 1 - x_4$ *и* $y_2 = 1 - x_5$ *даст систему с неотрицательными коэффициентами при линейных членах*

$$\begin{cases} x_1 + x_2 + x_3 + 2x_4 = 2 \\ x_1 + x_2 + 2x_3 + 3x_5 = 5 \end{cases}$$

*Исходное уравнение имеет лишь одно* $(0,1)$*-решение* $(1,1,0)^T$. *Новая система также имеет одно* $(0,1)$*-решение* $(1,1,0,0,1)^T$.

Если в исходном уравнении из условия теоремы 3 коэффициентами служат случайные целые числа независимо и равномерно распределенные на достаточно большом отрезке, то в новой системе некоторые коэффициенты детерминированы числом переменных и размером записи исходного уравнения, а другие коэффициенты почти равномерно распределены, каждый на своем отрезке от нуля до некоторого числа $p_k - 1$.

Наличие детерминированных коэффициентов приводит к тому, что мало эффективен алгоритм, предложенный Н.Н. Кузюриным [1]. В системе неравенств $A\mathbf{x} \leqslant \mathbf{b}$, соответствующей системе уравнений из теоремы 3, много допустимых $(0,1)$-решений.

Полученные результаты могут быть использованы для тестирования эвристических методов поиска $(0,1)$-решений систем уравнений.

## Список литературы

[1] *Кузюрин, Н. Н.* Полиномиальный в среднем алгоритм в целочисленном линейном программировании // Сибирский Журнал Исследования Операций. — 1994. — Т. 1, № 3. — С. 38–48.

[2] *Рыбалов, А. Н.* О генерической сложности проблемы о сумме подмножеств для полугрупп целочисленных матриц // Прикладная Дискретная Математика. — 2020. — № 50. — С. 118–126.

[3] *Селиверстов, А. В.* Двоичные решения для больших систем линейных уравнений // Прикладная Дискретная Математика. — 2021. — № 52. — С. 5–15.

[4] *Селиверстов, А. В.* О двоичных решениях систем уравнений // Прикладная Дискретная Математика. — 2019. — № 45. — С. 26–32.

[5] *Dusart, P.* Explicit estimates of some functions over primes // The Ramanujan Journal. — 2018. — V. 45. — P. 227–251.

### Библиографическая ссылка

*Селиверстов, А. В.* О сводимости систем линейных уравнений // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 262–266.
https://doi.org/10.26456/mfcsics-21-36

### Сведения об авторах

Селиверстов Александр Владиславович
Институт проблем передачи информации им. А. А. Харкевича РАН. Ведущий научный сотрудник

*Россия, 127051, Москва, Большой Каретный пер. 19, стр. 1*
*E-mail: slvstv@iitp.ru*

УДК 519.216
AMS MSC2020: 60G18

# Асимптотические оценки вероятности переполнения большого буфера телекоммуникационной системы для случая неоднородного входящего потока

**Сидорова О. И.**[*], **Суслов Л. В.**[**], **Хохлов Ю. С.**[**]

[*]Тверской государственный университет
[**]Московский государственный университет им. М. В. Ломоносова

Аннотация. В данной работе получена асимптотическая верхняя граница для самоподобного трафика, являющегося суммой $n$ независимых фрактальных броуновских движений с разными показателями Херста $H_1 < H_2 < \ldots < H_n$.

Ключевые слова: фрактальное броуновское движение, неоднородный трафик, вероятность переполнения буфера.

## Введение

Современный сетевой трафик имеет сложную структуру, что затрудняет перераспределение ресурсов для его эффективного обслуживания. Потери данных из-за переполнения буферов и увеличение времени задержки в трансляции негативно сказываются на качестве передаваемой видео и аудио-информации. Поэтому исследование влияния трафика на данные характеристики является важной задачей для сетевого конфигурирования.

Теоретические и практические исследования [5] показали, что фрактальное броуновское движение (ФБМ) адекватно описывает современный трафик. ФБМ — это самоподобный процесс с длинной памятью и легкими хвостами.

Оценка асимптотической нижней границы для вероятности переполнения конечного буфера в условиях ФБМ получена в работе [4].

В статьях [3, 5] приведены верхние границы для такой вероятности. В работе [6] описана асимптотика вероятности переполнения в условиях большого буфера.

В рамках данной статьи будет использован метод, предложенный в [1].

## 1. Основные определения

Определение 1. *Случайный процесс* $X = (X(t),\ t \geqslant 0)$ *называется самоподобным с параметром Херста* $H > 0$, *если он удовлетворяет условию*

$$X(t) \stackrel{d}{=} c^{-H} X(ct),\ \forall t \geqslant 0,\ \forall c > 0,$$

*где* $\stackrel{d}{=}$ *означает равенство конечномерных распределений.*

Определение 2. *Дробным броуновским движением с параметром* $H$ *называется гауссовский процесс* $(B_H(t), t \geqslant 0)$ *с нулевым средним и ковариационной функцией*

$$\gamma(t,s) = \frac{\sigma^2}{2}\Big(|t|^{2H} + |s|^{2H} - |t-s|^{2H}\Big), \quad 0 < H < 1. \tag{1}$$

*При* $H = 1/2$ *имеем обычное броуновское движение с корреляционной функцией*

$$\gamma(t,s) = \sigma^2 \min\{t,s\}.$$

## 2. Основные результаты

Рассмотрим систему массового обслуживания, на которую подается следующий входящий поток:

$$A(t) = mt + \sigma B_H(t), \tag{2}$$

где $(B_H(t),\ t \in R^1)$ — ФБМ с параметром $1/2 < H < 1$, $m$ — средняя интенсивность потока, $\sigma > 0$ — масштабный параметр.

В системе имеется одного устройство, с постоянной скоростью обслуживания $C > 0$. Тогда интенсивность трафика равна $r =$

$C - m > 0$ и у процесса нагрузки $Q(t) = \sup\limits_{t \geqslant s} (A(t) - A(s) - C(t-s))$ существует стационарное распределение

$$Q \stackrel{d}{=} \sup_{t \geqslant 0} (A(t) - Ct). \tag{3}$$

Нас интересует вероятность переполнения

$$\varepsilon(b) := P[Q > b], \tag{4}$$

где $b > 0$ — некоторый пороговый уровень, например, размер буфера.

Обозначим через $B_{1/2}(t)$ — стандартное броуновское движение и положим $\gamma = (2H-2)^{-1}$, $b^* = b \cdot (\sigma/r^H)^{2\gamma}$. В силу автомодельности процесса $B_H(t)$, неравенства Слепяна и теоремы 3.1 из [1] справедливо

$$P\left(\sup_{t\geqslant 0}\{\sigma B_H(t) - rt\} > b\right) = P\left(\sup_{t\geqslant 0}\{B(t) - t\} > b^*\right) \leqslant$$

$$\leqslant P\left(\sup_{t\geqslant 0}\{B_{1/2}(t^{2H}) - t\} > b^*\right) \leqslant \int\limits_0^\infty \frac{1}{\sqrt{2\pi t^{2H}}} \exp\left(-\frac{(t+b^*)^2}{2t^{2H}}\right) dt \sim$$

$$\sim \sqrt{\frac{H}{1-H}} \cdot e^{-c(H,\sigma,r)\cdot b^{2-2H}}, \quad b \to \infty, \tag{5}$$

где $c(H, \sigma, r)$ есть некоторая явно вычисляемая константа, а $\sim$ означает асимптотику.

Пусть теперь

$$A(t) = mt + \sum_{i=1}^n \sigma_i B_i(t) = (m_1 + \ldots + m_n)t + \sum_{i=1}^n \sigma_i B_i(t), \tag{6}$$

где $m_i$, $\sigma_i > 0$, $i = \overline{1,n}$ — средние интенсивности и масштабные параметры потоков, $(B_i(t) = B_{H_i}(t),\ t \in R^1)$ — независимые ФБМ с параметрами $1/2 < H_1 < \ldots < H_n < 1$.

Воспользуемся методом разделения потоков (de-multiplexing) трафика на отдельные очереди с $b_i = b/n$, $r_i = C_i - m_i$, $\overline{1,n}$, $C = C_1 + \ldots + C_n$. Имеем

$$P[Q > b] \leqslant \sum_{i=1}^n P\left(\sup_{t\geqslant 0}\{\sigma_i B_i(t) - r_i t\} > b_i\right). \tag{7}$$

Теорема 1. *В рамках описанной выше неоднородной модели верхняя асимптотическая граница для вероятности переполнения имеет следующий вид:*

$$P[Q > b] \leqslant \sqrt{\frac{H_n}{1 - H_n}} \cdot e^{-c(H_n, \sigma, r_n) \cdot b^{2-2H_n}}, \quad b \to \infty, \tag{8}$$

*где*

$$c(H, \sigma, r) = \frac{r^{2H}}{H^{2H} n^{2-2H} (1 - H)^{2-2H} \sigma^2}.$$

**Заключение**

В данной работе получена асимптотическая верхняя граница для вероятности переполнения большого буфера в условиях неоднородного самоподобного входящего потока, являющегося суммой независимых фрактальных броуновских движений. Данная оценка имеет простое аналитическое выражение, что облегчает ее практическое применение. Полученные результаты могут быть полезны при разработке механизмов управления и контроля для современных высокоскоростных сетей телекоммуникации.

**Список литературы**

[1] *Dębicki, K.* On the supremum from Gaussian processes over infinite horizon / K. Dębicki, Z. Michna, T. Rolski // Probability and Mathematical Statistics. — 2001. — Vol. 18. — P. 83–100.

[2] Is network traffic approximated by stable Levy motion or fractional Brownian motion? / Th. Mikosch, S. Resnick, H. Rootzen, A. Stegeman // The Annals of Applied Probability. — 2002. — Vol. 12, №1. — P. 23–68.

[3] *Lukashenko, O. V.* On the overflow probability asymptotics in a Gaussian queue / O. V. Lukashenko, E. V. Morozov, M. Pagano // Informatics and Applications. — 2014. — Vol. 8, №2. — P. 28–38.

[4] *Norros, I.* A Storage Model with Self-Similar Input // Queueing Systems. — 1994. — Vol. 16. — P. 387–396.

[5] *Rizk, A.* Sample path bounds for long memory fbm traffic / A. Rizk, M. Fidler // 29th Conference on Information Communications,

INFOCOM'10 Proceedings. — Piscataway, NJ, USA : IEEE Press, 2010. — P. 61–65.

[6] *Питербарг, В. И.* Двадцать лекций о гауссовских процессах. — М. : МЦНМО, 2015. — 188 с.

### Библиографическая ссылка

*Сидорова, О. И.* Асимптотические оценки вероятности переполнения большого буфера телекоммуникационной системы для случая неоднородного входящего потока / О. И. Сидорова, Л. В. Суслов, Ю. С. Хохлов // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 267–271. https://doi.org/10.26456/mfcsics-21-37

### Сведения об авторах

1. Оксана Игоревна Сидорова
Тверской государственный университет. Доцент

*Россия, 170002, г. Тверь, Садовый переулок, д. 35*
*E-mail: oksana.i.sidorova@yandex.ru*

2. Лев Владимирович Суслов
Московский государственный университет им. М. В. Ломоносова. Аспирант

*Россия, 119991, г. Москва, ГСП-1, Воробьевы горы, д. 1, стр. 52-2, ВМК МГУ им. М. В. Ломоносова*
*E-mail: suslov.lev@mail.ru*

3. Юрий Степанович Хохлов
Московский государственный университет им. М. В. Ломоносова. Профессор

*Россия, 119991, г. Москва, ГСП-1, Воробьевы горы, д. 1, стр. 52-2, ВМК МГУ им. М. В. Ломоносова*
*E-mail: yskhokhlov@yandex.ru*

УДК 510.644
AMS MSC2020: 03C13

# In defense of the self-reference quantifier S$x$. Approximation by dynamic systems.

**Stepanov V. A.**

Dorodnitsyn Computing Center of FIC CSC RAS

Abstract. Arguments in defense of introducing the self-referencing quantifier S$x$ and its approximation on dynamical systems are consistently presented. The case of classical logic is described in detail. Generated 3-valued truth tables that match the corresponding Priest tables [5]. In the process of constructing 4-valued truth tables, two more truth values were revealed that did not coincide with the original ones. Therefore, the closed tables turned out to be 6-valued. De Morgan's law confirmed in 6-valued truth tables.

Keywords: self-reference quantifier S$x$, dynamic systems, truth table, Liar, TruthTeller.

## Introduction

We are talking about the S icon, which first appeared in the article [4]: $Q =_{df} S_Q P$. According to the meaning, S indicates that the entire expression belongs to self-referencing, and introduces the entire self-referential construction to the rank of WFF. The Liar sentence: $S_Q \sim TQ$.

## 1. Basic definitions

Self-referential sentences deserve to be marked out in language for their self-referencing. To do this, we fix the self-referencing of the sentence using a special icon — the self-referencing icon S$x$, which is placed in front of the predicate $P(x)$, which we call the core of the self-referential sentence. As a result, a self-referential sentence looks like this:

$$\mathrm{S}xP(x). \tag{1}$$

In place of the variable $x$ in $P(x)$ from (1), nothing can be substituted except for this sentence itself. You cannot substitute anything in the newly received sentence in $x$, except for this sentence itself, etc. Those. a self-referential sentence is outwardly closed, and the expression S$x$, according to this criterion, can well be attributed to quantifiers, because it is the presence of S$x$ that makes expression (1) closed. Expression (1) obeys the axiom of self-reference, which is the essence of the axiom of a fixed point, [3]:

$$\mathrm{S}xP(x) = P(\mathrm{S}xP(x)) \tag{2}$$

Peirce [2] intuitively applied (2) to generate an infinite Liar sentence:

$$\mathrm{S}xP(x) = P(P(P(P(\ldots \mathrm{S}xP(x) \ldots)))) \tag{3}$$

This infinite sentence consists of an infinite number of nested Liar kernels. Let's break it down into iterative steps, discarding the "last" expression $\ldots \mathrm{S}xP(x) \ldots$

$$\mathrm{S}xP(x) \approx \mathsf{S}xP(x) \rightleftharpoons \langle x, P(x), P(P(x)), P(P(P(x))), \ldots \rangle \tag{4}$$

The $\approx$ indicates an approximation. Expression $\mathsf{S}xP(x)$ in (4) on the right will be considered as an approximation of a real self-referential sentence $\mathrm{S}xP(x)$. To denote the result of the approximation, we will choose the sign $\mathsf{S}$ to distinguish it from S — a real quantifier of self-reference. The expression $\mathsf{S}x$ will also be called a self-referencing quantifier, if this does not lead to an error.

In front of the sequence of kernels in (4), we insert the variable $x = P^0(x)$ to distinguish one specific branch of the approximation from another. Expression (4) is the definition of the trajectory of a dynamical system of the form $(\{0,1\}, P(x))$ with orbits $\langle P^n(x), n \in Z^+ \rangle$, where $P^n(x) = P(P^{n-1}(x))$. This justifies the title of our article. Expression (4) in the theory of dynamical systems [1] is called the trajectory or orbit of the dynamical system. We use the characteristics of such a movement here. Consider the case when the kernels of self-referential sentences $P(x)$ are composed of $Tr(x)$ using propositional connectives $\leftrightarrow$ and $\neg$:

$$P(x) \in \{Tr(x), \neg Tr(x), Tr(x) \leftrightarrow Tr(x), Tr(x) \leftrightarrow \neg Tr(x)\}. \tag{5}$$

The rest of the formulas we are considering are equivalent to these four. The variable$x$ and the predicates $P(x)$ from (5) in our case take

values from $\{0, 1\}$. It is easy to see that expression (4) is periodic, with a maximum period of 2. This means that the second and third terms of the sequence (4) determine the entire remaining infinite sequence. Therefore, in our case, we rightfully shorten the definition of a self-referencing quantifier as follows:

$$\mathsf{S}xP(x) \rightleftharpoons \langle x, P(x), P(P(x))\rangle. \tag{6}$$

Since there are only two values of $x$ in sequence (6) in our case: $x \in \{0, 1\}$, then statement (6) itself splits into two sequences. And since we have no reason to give preference to any one of them, we will combine them as equal rights elements of the set in (7):

$$\mathsf{S}xP(x) = \{\langle 1, P(1), P(P(1))\rangle, \langle 0, P(0), P(P(0))\rangle\}. \tag{7}$$

In the case when the values of $x$ will be more (or less) than two, the number of members of the sets in (6) and (7) should be changed accordingly. This is one of the properties of the definition of the approximation of the self-referencing quantifier $\mathsf{S}x$, which allows it to be used in other logical systems, and not only in classical ones, as in the case under consideration. Now let us define the action of the external negation sign $\neg$. To do this, we will divide our manipulations into several cases. The first of them is when the kernel $P(x)$ of a self-referential sentence is the identically true:
$[P(x) = (Tr(x) \leftrightarrow Tr(x))$, i. e. $P(0) = P(1) = 1]$ or the identically false: $[P(x) = (Tr(x) \leftrightarrow \neg Tr(x))$, i. e. $P(0) = P(1) = 0]$ formula. Then, for example, for $P(x) = 1$ we get

$$\begin{aligned} \neg\mathsf{S}xP(x) &= \neg\{\langle 1, 1, 1\rangle, \langle 0, 1, 1\rangle\} && (= \neg T) \\ &= \{\neg\langle 1, 1, 1\rangle, \neg\langle 0, 1, 1\rangle\} \\ &= \{\langle \neg 1, \neg 1, \neg 1\rangle, \langle \neg 0, \neg 1, \neg 1\rangle\} \\ &= \{\langle 0, 0, 0\rangle, \langle 1, 0, 0\rangle\} && (= F). \end{aligned}$$

In the case of nonidentical formulas, $Tr(x)$ (TruthTeller) or $\neg Tr(x)$ (Liar), the estimate of the formula changes along with the estimate for the free variable $x$:

$$\begin{aligned} \neg\mathsf{S}xP(x) &= \neg\{\langle 1, 0, 1\rangle, \langle 0, 1, 0\rangle\} \\ &= \{\langle \neg 1, \neg 0, \neg 1\rangle, \langle \neg 0, \neg 1, \neg 0\rangle\} \\ &= \{\langle 0, 1, 0\rangle, \langle 1, 0, 1\rangle\} \\ &= \{\langle 1, 0, 1\rangle, \langle 0, 1, 0\rangle\} && (= \mathsf{S}xP(x)) \end{aligned}$$

This is the table for the negation symbol:

| $\mathsf{S}xP(x)$ | $\neg\mathsf{S}xP(x)$ |
|---|---|
| $\{\langle 1,1,1\rangle;\langle 0,1,1\rangle\}=T$ | $F=\{\langle 1,0,0\rangle;\langle 0,0,0\rangle\}$ (False) |
| $\{\langle 1,0,1\rangle;\langle 0,1,0\rangle\}=A$ | $A=\{\langle 0,1,0\rangle;\langle 1,0,1\rangle\}$ (Antinomy) |
| $\{\langle 1,1,1\rangle;\langle 0,0,0\rangle\}=V$ | $V=\{\langle 0,0,0\rangle;\langle 1,1,1\rangle\}$ (Void) |
| $\{\langle 1,0,0\rangle;\langle 0,0,0\rangle\}=F$ | $T=\{\langle 1,0,0\rangle;\langle 0,0,0\rangle\}$ (True) |

We define two-place connectives $\circ\in\{\wedge,\vee,\to,\leftarrow,\leftrightarrow\}$ for two S-formulas $\mathsf{S}xP(x)$ and $\mathsf{S}xQ(x)$. We study such a variant of two-place connectives, when the trajectories of estimates of the formula $\mathsf{S}xP(x)$ of the one branch ($x=1$ or $x=0$) interact with the trajectories of the formula $\mathsf{S}xQ(x)$ of the same branch ($x=1$ or $x=0$):

$$\langle 1,P(1),P(P(1))\rangle\circ\langle 1,Q(1),Q(Q(1))\rangle,$$
$$\langle 0,P(0),P(P(0)\rangle\circ\langle 0,Q(0),Q(Q(0))\rangle:$$

$$\begin{gathered}\mathsf{S}xP(x)\circ\mathsf{S}xQ(x)=\\ \{\langle 1,P(1),P(P(1))\rangle,\langle 0,P(0),P(P(0)\rangle\}\circ\\ \{\langle 1,Q(1),Q(Q(1))\rangle,\langle 0,Q(0),Q(Q(0))\rangle\}=\\ \{\langle 1,P(1),P(P(1))\rangle\circ\langle 1,Q(1),Q(Q(1))\rangle,\\ \langle 0,P(0),P(P(0))\rangle\circ\langle 0,Q(0),Q(Q(0))\rangle\}=\\ \{\langle 1\circ 1,P(1)\circ Q(1),P(P(1))\circ Q(Q(1))\rangle,\\ \langle 0\circ 0,P(0)\circ Q(0),P(P(0))\circ Q(Q(0))\rangle\}.\end{gathered}$$

Here are examples of the interactions between the estimates of *Liar* A (and *TruthTeller* V) with T, F:

$$\begin{aligned}\mathrm{V}\wedge V&=\{\langle 1,1,1\rangle,\langle 0,1,1\rangle\}\wedge\{\langle 1,1,1\rangle,\langle 0,0,0\rangle\}=\\&=\{\langle 1,1,1\rangle,\langle 0,0,0\rangle\}=\mathrm{V}\\ \mathrm{A}\wedge A&=\{\langle 1,1,1\rangle,\langle 0,1,1\rangle\}\wedge\{\langle 1,0,1\rangle,\langle 0,1,0\rangle\}=\\&=\{\langle 1,0,1\rangle,\langle 0,1,0\rangle\}=\mathrm{A}\\ \mathrm{F}\wedge V&=\{\langle 1,0,0\rangle,\langle 0,0,0\rangle\}\wedge\{\langle 1,1,1\rangle,\langle 0,0,0\rangle\}=\\&=\{\langle 1,0,0\rangle,\langle 0,0,0\rangle\}=\mathrm{F}\\ \mathrm{F}\wedge A&=\{\langle 1,0,0\rangle,\langle 0,0,0\rangle\}\wedge\{\langle 1,0,1\rangle,\langle 0,1,0\rangle\}=\\&=\{\langle 1,0,0\rangle,\langle 0,0,0\rangle\}=\mathrm{F}\end{aligned}$$

## 2. Main results

Let's reproduce Priest's tables and compare them with ours, built on our rules: V and A

Hypothesis: p = V

| $\wedge$ | T | V | F |
|---|---|---|---|
| T | T | V | F |
| V | V | V | F |
| F | F | F | F |

Priest p

| $\wedge$ | t | p | f |
|---|---|---|---|
| t | t | p | f |
| p | p | p | f |
| f | f | f | f |

Hypothesis: p = A

| $\wedge$ | T | A | F |
|---|---|---|---|
| T | T | A | F |
| A | A | A | F |
| F | F | F | F |

Comparing our table for A (*Liar*) with Priest's table for p (*Liar*) in [5], we notice that they are identical. It should be borne in mind that our tables are built on a completely different principle, different from the principles of Priest's construction. And this inspires a certain optimism, when two completely different principles of construction, so to speak, "external" (priest's) and "internal" (ours), lead to the same result. Comparing our table for V (*TruthTeller*), with Priest's table for p (*Liar*) in [5], we notice that they have the same configuration. But Priest, in his work [5], considers only the *Liar* sentence. Therefore, we will build four-valued tables in which our A and V will be able to interact, with their different truth estimates.

| $\wedge$ | T | A | V | F |
|---|---|---|---|---|
| T | T | A | V | F |
| A | A | A | av | F |
| V | V | av | V | F |
| F | F | F | F | F |

| $\vee$ | T | A | V | F |
|---|---|---|---|---|
| T | T | T | T | T |
| A | T | A | va | A |
| V | T | va | V | V |
| F | T | A | V | F |

Here new assessments from interaction appear A and V: va and av.

$$A \wedge V = \{\langle 1,0,1\rangle, \langle 0,0,0\rangle\} = av$$
$$A \vee V = \{\langle 1,1,1\rangle, \langle 0,1,0\rangle\} = va$$

Closed value tables will look like this:

| $\wedge$ | T | av | A | V | va | F |
|---|---|---|---|---|---|---|
| T | T | av | A | V | va | F |
| av | av | av | av | av | av | F |
| A | A | av | A | av | A | F |
| V | V | av | av | V | V | F |
| va | va | av | A | V | va | F |
| F | F | F | F | F | F | F |

| $\vee$ | T | av | A | V | va | F |
|---|---|---|---|---|---|---|
| T | T | T | T | T | T | T |
| av | T | av | A | V | va | av |
| A | T | A | A | va | va | A |
| V | T | V | va | V | va | V |
| va | T | va | va | va | va | va |
| F | T | av | A | V | va | F |

This allows us to prove the following lemma:

Lemma 1. 1) *The sentences Liar (A) have the tabular model, coinciding with tabular model Liar (p) of Priest [5] and, accordingly, the same evidential theory.*

2) *The sentences TruthTeller (V) have the same configuration tabular model, coinciding with configuration tabular model Liar (p) of Priest [5].*

3) *When constructing truth tables for the interaction of V and A, new truth values were obtained: V∧A = av and V∨A = va; ¬av=va; ¬va=av.*

In the same way, we will construct tables for disjunction, implication and reverse implication, using the latter two and conjunction to construct the equivalence.

## Conclusion

The described form of constructing a model of the logic of self-referential sentences reduces the many-valued of estimates to a two-valued logic, which corresponds to the spirit of Suzsko's sentences.

## References

[1] Dynamics of one-dimensional mappings / A. N. Sharkovskii, S. F. Kolyada, A. G. Sivak, V. V. Fedorenko. — Kiev : Naukova Dumka, 1989. — 216 p. [In Russian]

[2] *Emily, M.* Pierce's Paradoxical Solution to the Liar's Paradox // Notre Dame Journal of Formal Logic. — 1975. — Vol. 16, №3. — P. 369–374.

[3] *Feferman, S.* Toward Useful Type-Free Theories I. // The Journal of Symbolic Logic. — 1984. — Vol. 49, №1. — P. 75–111.

[4] *Johnstone, A.* Self-reference, the Double Life and Gödel // Logique et Analyse. — 1981. — Vol. 24. — P. 35–47.

[5] *Priest, G.* The Logic of Paradox // Journal of Philosophical Logic. — 1979. — Vol. 8. — P. 219–241.

**Библиографическая ссылка**

*Stepanov, V. A.* In defense of the self-reference quantifier S$x$. Approximation by dynamic systems. // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 272–278.
https://doi.org/10.26456/mfcsics-21-38

**Сведения об авторах**

Vladimir Alekseevich Stepanov
Dorodnitsyn Computing Center of FIC CSC RAS. Researcher

*Vavilov st. 40, 119333 Moscow, Russia*
*E-mail: vastvast@yandex.ru*

УДК 510.643
AMS MSC2020: 03B45

# О полноте модальных предикатных логик в семантике Крипке

**Шехтман В. Б.**

Институт проблем передачи информации им. А. А. Харкевича РАН;
НИУ «Высшая школа экономики»;
Московский государственный университет им. М. В. Ломоносова

Аннотация. В работе дан обзор известных и новых результатов о полноте и неполноте в семантике Крипке для минимальных предикатных расширений пропозициональных модальных логик.

Ключевые слова: модальная логика, логика предикатов, модель Крипке, полнота по Крипке.

## Введение

Около 50 лет назад было замечено, что, в отличие от логик высказываний, многие модальные логики предикатов неполны в семантике Крипке. Хотя имеются общие теоремы о полноте о предикатных логиках с постоянными областями (см. [3]), для логик с расширяющимися областями подобные результаты неизвестны. В данной работе рассматриваются логики вида $\mathbf{Q\Lambda}$ — минимальные предикатные расширения пропозициональных логик $\mathbf{\Lambda}$; уже для них проблема полноты нетривиальна. Мы даем обзор состояния дел по этой теме.

## 1. Основные определения

Как и в [3], модальные предикатные формулы рассматриваются в сигнатуре без равенства со счетным множеством предикатных символов каждой из валентностей $0, 1, \ldots$; базовые связки: $\bot, \to, \Box$. Модальная предикатная логика — множество формул, содержащее аксиомы модальной пропозициональной логики $\mathbf{K}$, аксиомы классического исчисления предикатов и замкнутое относительно

правил Modus Ponens, $A/\forall xA$, $A/\Box A$ и предикатной подстановки. $\mathbf{Q\Lambda}$ — минимальное предикатное расширение пропозициональной логики $\mathbf{\Lambda}$.

Предикатная шкала Крипке над пропозициональной шкалой $F = (W, R)$ — это пара $\mathbf{F} = (F, D)$, где $D = (D_u)_{u\in W}$, все области $D_u$ непусты и $D_u \subseteq D_v$ при $uRv$. Оценка $\xi$ на $\mathbf{F}$ — функция, переводящая каждый $n$-местный предикатный символ $P^n_k$ в семейство $n$-местных отношений на $D_u$, то есть $\xi(P^n_k) = (\xi_u(P^n_k))_{u\in W}$, где $\xi_u(P^n_k) \subseteq D^n_u$ для $n \neq 0$ и $\xi_u(P^0_k) \in \{0, 1\}$. Пара $M = (\mathbf{F}, \xi)$ — модель Крипке над $\mathbf{F}$.

Истинность $D_u$-предложения (замкнутой формулы с константами из $D_u$) в точке $u$ из $M$ определяется стандартно, по рекурсии. В частности,

$$M, u \vDash P^n_k(a_1, \dots, a_n) \Leftrightarrow (a_1, \dots, a_n) \in \xi_u(P^n_k),$$
$$M, u \vDash P^0_k \Leftrightarrow \xi_u(P^0_k) = 1,$$
$$M, u \vDash \forall xA(x) \Leftrightarrow \forall a \in D_u\ M, u \vDash A(a),$$
$$M, u \vDash \Box A \Leftrightarrow \forall v \in R(u)\ M, v \vDash A.$$

Формула $A(\mathbf{x})$ общезначима на $\mathbf{F}$, если ее универсальное замыкание $\forall \mathbf{x}A(\mathbf{x})$ истинно во всех точках всех моделей Крипке над $\mathbf{F}$. Модальной логикой класса шкал $\mathcal{C}$ называется множество всех формул, общезначимых на всех шкалах из $\mathcal{C}$. Такие логики называются полными по Крипке.

Данная работа посвящена проблеме полноты для логик вида $\mathbf{Q\Lambda}$. Перечислим пропозициональные логики и аксиомы, которые здесь упоминаются.

$$\mathbf{K4} = \mathbf{K} + \Box p \to \Box\Box p, \qquad \mathbf{S4} = \mathbf{K4} + \Box p \to p,$$
$$\Box\cdot\mathbf{T} = \mathbf{K} + \Box(\Box p \to p), \qquad \mathbf{SL4} = \mathbf{K4} + \Box p \leftrightarrow \Diamond p,$$
$$\mathbf{K4.2} = \mathbf{K4} + \Diamond\Box p \to \Box\Diamond p, \quad \mathbf{S4.2} = \mathbf{K4.2} + \Box p \to p,$$
$$\mathbf{K4.3} = \mathbf{K4} + \Box(p \wedge \Box p \to q) \vee \Box(q \wedge \Box q \to p),$$
$$\mathbf{S4.3} = \mathbf{K4.3} + \Box p \to p, \qquad Ad_n = \bigwedge_{i=1}^{n} \Diamond p_i \to \Diamond(\bigwedge_{i=1}^{n} \Diamond p_i),$$
$$\mathbf{GL} = \mathbf{K} + \Box(\Box p \to p) \to \Box p, \quad \mathbf{K05} = \mathbf{K} + \Diamond\Diamond p \to \Box\Diamond p,$$

$$\mathbf{Alt_n} = \mathbf{K} + \bigwedge_{i=1}^{n+1} \Diamond p_i \to \bigvee_{1 \leqslant i < j \leqslant n+1} \Diamond(p_i \wedge p_j).$$

Шкалы Крипке для **K05** задаются условием «густоты»: $R^{-1} \circ R^2 \subseteq R$, для $\mathbf{Alt_n}$ — условием ограниченного ветвления: $\forall x\, |R(x)| \leqslant n$, для $\mathbf{K} + Ad_n$ — условием $n$-плотности: $\bigcap_{i=1}^{n} R(x_i) \subseteq R^{-1}(\bigcap_{i=1}^{n} R(x_i))$.

## 2. Теоремы о неполноте по Крипке

Теорема 1 (см. [4]). *Для* $\mathbf{\Lambda} \supseteq \mathbf{S4}$, *логика* $\mathbf{Q\Lambda}$ *может быть полной только если* $\mathbf{\Lambda} \supseteq \mathbf{S5}$ *или* $\mathbf{\Lambda} \subseteq \mathbf{S4.3}$.

Теорема 2 (см. [7]). *Если* $\Box \cdot \mathbf{T} \subseteq \mathbf{\Lambda} \subseteq \mathbf{SL4}$, *то логика* $\mathbf{Q\Lambda}$ *неполна.*

Теорема 3 (см. [5]). *Логика* **QGL** *неполна.*

## 3. Теоремы о полноте по Крипке

Определение 1. *Односторонняя PTC-логика — это модальная пропозициональная логика, которая аксиоматизируется формулами вида* $\Box p \to \Box^n p$, *а также замкнутыми пропозициональными формулами.*

Теорема 4 (см. [3]). *Если* $\mathbf{\Lambda}$ *— односторонняя PTC-логика, то* $\mathbf{Q\Lambda}$ *полна.*

Теорема 5 (см. [1], [3]). *Логика* **QS4.2** *полна.*

Теорема 6 (см. [2]). *Логики* **QS4.3**, **QK4.3**, $\mathbf{QK4.3} + Ad + \Diamond\top$ *полны.*

Теорема 7 (см. [6]). $\mathbf{Q\Lambda}$ *полна для следующих логик* $\mathbf{\Lambda}$*:* $\mathbf{K4} + Ad_n$, $\mathbf{K} + Ad_n$, **K4.2**, $\mathbf{K4.2} + Ad_n$.

Теорема 8 (см. [8]). *Логика* $\mathbf{QAlt_n}$ *полна.*

Теорема 9. *Логика* $\mathbf{QK05} + \Box^3\bot$ *полна.*

Для доказательства двух последних теорем применяются «квазиканонические модели». Напомним сначала определение канонических моделей [3].

Определение 2. *Зафиксируем счетное множество констант* $S^*$. *L-плей с — это максимальная L-непротиворечивая теория Хенкина в сигнатуре с дополнительными константами из некоторого подмножества* $S^*$ *с бесконечным дополнением.*

*Каноническая модель $VM_L$ модальной предикатной логики $L$ имеет вид $(VP_L, R_L, D_L, \xi_L)$, где*

- *$VP_L$ — множество всех $L$-плейсов,*
- *$\Gamma R_L \Delta$, если для всех $A$ из $\Box A \in \Gamma$ следует $A \in \Delta$,*
- *$(D_L)_\Gamma$ (обозначается $D_\Gamma$) — множество всех констант из $\Gamma$,*
- *$VM_L, \Gamma \vDash A$ тогда и только тогда, когда $A \in \Gamma$ для атомарных $D_\Gamma$-предложений $A$.*

Теорема 10 (О канонической модели). *Для любого $L$-плейса $\Gamma$ и $D_\Gamma$-предложения $A$, $VM_L, \Gamma \vDash A$ тогда и только тогда, когда $A \in \Gamma$.*

Следующие определения несколько отличаются от [3].

Определение 3. *Модель Крипке $M' = (W', R', D', \xi')$ называется слабой селективной подмоделью модели Крипке $M = (W, R, D, \xi)$, если*

- *$W' \subseteq W$; $R' \subseteq R$; для всех $w \in W'$, $D_w = D'_w$; $\xi'_w = \xi_w$,*
- *для всех $D_w$-предложений $A$,*

$$M, w \not\vDash \Box A \Longrightarrow \exists u \in R'(w)\, M, u \not\vDash A.$$

Определение 4. *Квазиканоническая модель логики $L$ — это слабая селективная подмодель ее канонической модели.*

Теорема 11. *Для любого $L$-плейса $\Gamma$ из квазиканонической модели $M$ и $D_\Gamma$-предложения $A$ $M, \Gamma \vDash A \Leftrightarrow A \in \Gamma$.*

Для доказательства теоремы 8 квазиканоническая модель, содержащая данный $L$-плейс, последовательно строится следующим образом. Если уже построено $\Gamma$, то для каждой формулы $\Diamond A \in \Gamma$ добавляется $L$-плейс $\Delta \in R_L(\Gamma)$, содержащий $A$ (если такого нет среди уже построенных). $\mathbf{Alt_n}$ обеспечивает ограниченное ветвление.

Для доказательства теоремы 9 квазиканоническая модель, содержащая данный $L$-плейс $\Gamma$, строится в 2 этапа. Пусть $\{\Diamond A_1, \Diamond A_2, \ldots\}$ — список всех формул из $\Gamma$, начинающихся с $\Diamond$. На первом этапе для каждой формулы $\Diamond_i A$ добавляется $L$-плейс $\Delta_i \in R_L(\Gamma)$, содержащий $A$ (если такого нет среди $\Gamma, \Delta_1, \ldots, \Delta_{i-1}$). При этом $L$-плейсы $\Delta_i$ должны быть разделены, то есть $D_{\Delta_i} \cap D_{\Delta_j} = D_\Gamma$ при $i \neq j$; это условие всегда можно обеспечить в конструкции Хенкина.

На втором этапе для каждого $i$ и формулы $\Diamond B \in \Delta_i$ строится содержащий ее $L$-плейс из $\bigcap_j R_L(\Delta_j)$. Такой плейс всегда существует, благодаря непротиворечивости теорий

$$\{B\} \cup \bigcup_j \{A \mid \Box A \in \Delta_j\}.$$

Непротиворечивость устанавливается с использованием аксиомы **K05** и разделенности теорий $\Delta_j$.

### Заключение

Таким образом, для логик $\mathbf{Q\Lambda}$ сохраняется прежняя картина: полные логики — островки в океане неполных. Во многих случаях вопрос о полноте открыт: например, для $\mathbf{\Lambda} = \mathbf{K05} + \Box^4\bot$ или когда $\mathbf{\Lambda}$ — логика конечного дерева (со строгим порядком).

### Список литературы


[1] *Corsi, G.* Directed frames / G. Corsi, S. Ghilardi // Archive for Mathematical Logic. — 1989. — Vol. 29. — P. 53–67.

[2] *Corsi, G.* Quantified modal logics of positive rational numbers and some related systems // Notre Dame Journal of Formal Logic. — 1993. — Vol. 34. — P. 263–283.

[3] *Gabbay, D.* Quantification in nonclassical logic, v. 1. / D. Gabbay, V. Shehtman, D. Skvortsov. — Amsterdam : Elsevier, 2009. — 615 p.

[4] *Ghilardi, S.* Incompleteness results in Kripke semantics // Journal of Symbolic Logic. — 1991. — Vol. 56, №2. — P. 517–538.

[5] *Montagna, F.* The predicate modal logic of provability // Notre Dame J. of Formal Logic. — 1984. — Vol. 25, №2. — P. 179–189.

[6] *Shehtman, V.* On Kripke completeness of some modal predicate logics with the density axiom // Advances in Modal Logic. — College Publications, 2018. — Vol. 12. — P. 559–576.

[7] *Shehtman, V.* Kripke completeness of modal predicate logics around quantified **K5**. — 2021, submitted.

[8] *Shehtman, V.* Semiproducts, products, and modal predicate logics / V. Shehtman, D. Shkatov. — In preparation.

### Библиографическая ссылка

*Шехтман, В. Б.* О полноте модальных предикатных логик в семантике Крипке // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 279–284.
https://doi.org/10.26456/mfcsics-21-39

### Сведения об авторах

Валентин Борисович Шехтман
Институт проблем передачи информации им. А. А. Харкевича РАН;
НИУ «Высшая школа экономики»;
Московский государственный университет им. М. В. Ломоносова.
Главный научный сотрудник; профессор

*Россия, 127051, Москва, Б. Каретный, 19 стр. 1*
*E-mail: vshehtman@gmail.com*

УДК 004.773
AMS MSC2020: 94D99

# Методические рекомендации по оптимизации параметров системы аутентификации на основе использования универсальных хэш-функций и случайных цепочек бит

**Яковлев В. А.**[*], **Савинова С. А.**[**], **Гатчин Ю. А.**[**], **Поляков В. И.**[**], **Чикалов Н. В.**[**]

[*]Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
[**]Университет ИТМО

Аннотация. В статьях [3,4] предлагается метод аутентификации на основе использования универсальных хэш-функций и случайных цепочек бит. Настоящая статья является продолжением и содержит методические рекомендации по оптимизации параметров предложенной системы аутентификации.

Ключевые слова: хэш-функции, случайные цепочки бит, аутентификация, оптимизация.

## Введение

Разработанный способ аутентификации ключа [3, 4], распределяемого методом Диффи – Хеллмана, требует методики оптимизации параметров.

В предложенном ранее способе аргументы функций являются дискретными, иными словами — в результате испытания принимают значения с определенными вероятностями. Поэтому для оптимизации используется метод дихотомии и метод перебора [1].

Исходными данными в такой системе аутентификации являются: длина ключа $n_0$, вероятность отличия бит в аутентифицирующих

последовательностях $p_m$, требуемая вероятность ложного отклонения [2] $P_f < P_f^{\text{треб}}$, требуемая вероятность ложной аутентификации [2] $P_d < P_d^{\text{треб}}$.

Необходимо определить следующие оптимальные параметры: длину подблока аутентификации $m$, длину аутентификатора $v$.

При выборе определяемых параметров необходимо обеспечить требуемые вероятность ложного отклонения аутентификации и вероятность ложной аутентификации, а также минимизировать общую длину аутентификаторов всех блоков сообщения.

## 1. Основные определения

Определение 1. *Аутентификация — это процедура проверки подлинности.*

*Хэш-функция — это функция, реализующая определенный алгоритм и выполняющая преобразование массива входных данных произвольной длины в битовую строку фиксированной длины.*

*Случайные цепочки бит — это двоичные последовательности, которые пользователи получают по дополнительному каналу.*

*Оптимизация — это процесс максимизации выгодных характеристик.*

*Вероятность ложной аутентификации — это вероятность события, которое наступает, когда злоумышленник подменяет подблоки ДХ-значения и осуществляет подбор к ним аутентификаторов таким образом, что число неправильно подобранных аутентификаторов плюс число неправильно принятых блоков из-за несовпадения случайных цепочек бит не больше чем пороговое значение.*

*Вероятность ложного отклонения — это вероятность события, которое наступает, когда число неправильно аутентифицированных блоков больше установленного порогового значения из-за несогласованности случайных цепочек бит.*

## 2. Основные результаты

Удалось разработать методику, по которой удается определить оптимальные параметры системы, при которых расчетные значения ложного отклонения и ложной аутентификации не превышают $10^{-6}$.

Методика включает в себя следующие разделы:

1) Оценка величины возможного отличия ложного сообщения от истинного и оценка вероятности формирования такого сообщения.
2) Оценка максимально вероятного количества ложных подблоков в последовательности.
3) Построение зависимостей вероятности ложного отклонения последовательности в отсутствие навязывания для различных длин блоков и определение допустимого порога стирания.
4) Построение зависимости ложной аутентификации для различных длин подблоков и длине аутентификатора равной длине подблока.
5) Анализ значений вероятностей ложного отклонения и ложной аутентификации для различных значений порогового значения, длины подборка и длины аутентификатора.
6) Использование способа увеличения безошибочности аутентифицирующих последовательностей за счет использования процедуры помехоустойчивого кодирования.
7) Проведение оптимизации длины аутентификатора.
8) Расчет требуемой длины ключа для аутентификации всего сообщения.

В качестве доказательства применимости данной методики рассмотрен подробно пример ее использования и получения оптимальных параметров.

## Заключение

В работе разработана методика оптимизации системы аутентификации значений Диффи – Хеллмана. Подробно рассмотрен пример, который демонстрирует эффективность данной методики. Выбрав должным образом параметры системы аутентификации можно обеспечить малые значения вероятности ложной аутентификации и вероятности навязывания ложного ключа, при этом минимизировав суммарную длину всех аутентификаторов, что и говорит о достижении поставленной оптимизационной задачи.

### Список литературы

[1] *Гребенникова, И. В.* Методы оптимизации : учебное пособие. — Екатеринбург : УрФУ, 2017. — 148 с.

[2] *Феллер, В.* Введение в теорию вероятностей и ее приложения : в 2 томах. Т. 2. / Пер. с англ. Р. Л. Добродушина, А. А. Юшкевич, С. А. Молчанова. — 2-е изд. — Москва : Мир, 2017. — 748 с.

[3] *Яковлев, В. А.* Аутентификация сеансового ключа на основе универсальных хэш-функций и случайных цепочек бит / В. А. Яковлев, С. А. Савинова // i-methods — Информатика, вычислительная техника и управление. — 2020. — Т. 12, №4. — URL: http://intech-spb.com/wp-content/uploads/archive/2020/4/2_jakovlev_cavinova.pdf

[4] *Яковлев В. А.* Аутентификация ключей, распределяемых методом Диффи – Хеллмана, на основе использования универсальных хэш-функций и помехоустойчивого кодирования // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2019). Сборник научных статей VIII Международной научно-технической и научно-методической конференции. — 2019. — Т. 4, №2. — С. 762–767.

### Библиографическая ссылка

Методические рекомендации по оптимизации параметров системы аутентификации на основе использования универсальных хэш-функций и случайных цепочек бит / В. А. Яковлев, С. А. Савинова, Ю. А. Гатчин [et al.] // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем». Сборник трудов. — Тверь : ТвГУ, 2021. — С. 285–289. https://doi.org/10.26456/mfcsics-21-40

### Сведения об авторах

1. Виктор Алексеевич Яковлев
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича. Профессор

*Россия, 193232, проспект Большевиков д.22, к.1*
*E-mail: viyak@bk.ru*

2. Светлана Алексеевна Савинова
Университет ИТМО. Магистрант

*Россия, 197101, Кронверкский проспект, д.49, литер А.*
*E-mail: savinova-sveta@mail.ru*

3. Юрий Арменакович Гатчин
Университет ИТМО. Профессор

*Россия, 197101, Кронверкский проспект, д.49, литер А.*
*E-mail: gatchin1952@mail.ru*

4. Владимир Иванович Поляков
Университет ИТМО. Доцент

*Россия, 197101, Кронверкский проспект, д.49, литер А.*
*E-mail: v_i_polyakov@mail.com*

5. Никита Вячеславович Чикалов
Университет ИТМО. Магистрант

*Россия, 197101, Кронверкский проспект, д.49, литер А.*
*E-mail: nik.chikalow2011@yandex.ru*

Научное издание

Всероссийская научная конференция
«Математические основы информатики
и информационно-коммуникационных систем»

Сборник трудов

Тверь
3–8 декабря 2021 г.

Под редакцией С. М. Дудакова и Б. Н. Карлова

Подписано в печать 29.11.2021
Усл. п. л. 16,86. Тираж 300 экз.
Заказ № 366

Тверской государственный университет
Издательство Тверского государственного университета
Адрес: 170100, г. Тверь, Студенческий пер., 12, корпус Б
Тел.: (4822) 35-60-63