

2020

Volume 27

No 3

MODELING AND ANALYSIS OF INFORMATION SYSTEMS

SCIENTIFIC JOURNAL

Start date of publication — 1999

Published quarterly

FOUNDER

P.G. Demidov Yaroslavl State University

EDITORIAL OFFICE

14 Sovetskaya str., Yaroslavl 150003, Russian Federation

Website: <http://mais-journal.ru>

E-mail: mais@uniyar.ac.ru

Phone: +7 (4852) 79-77-73

2020

Том 27

№ 3

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ

НАУЧНЫЙ ЖУРНАЛ

Издается с 1999 года

Выходит 4 раза в год

УЧРЕДИТЕЛЬ

федеральное государственное бюджетное образовательное учреждение высшего образования
«Ярославский государственный университет им. П. Г. Демидова»

РЕДАКЦИЯ

ул. Советская, 14, Ярославль, 150003, Российская Федерация

Website: <http://mais-journal.ru>

E-mail: mais@uniyar.ac.ru

Телефон: +7 (4852) 79-77-73

Свидетельство о регистрации СМИ ПИ № ФС 77–66186 от 20.06.2016 выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций. Подписной индекс — 31907 в Объединенном каталоге «Пресса России». Корректор — М. С. Каряева, корректор английских текстов — М. С. Комар. Подписано в печать 18.09.2020. Дата выхода в свет 30.09.2020. Формат 200×265 мм. Объем 112 с. Тираж 46 экз. Свободная цена. Заказ 050/020. Адрес типографии: ул. Советская, 14, оф. 109, Ярославль, 150003, Россия. Адрес издателя: Ярославский государственный университет им. П. Г. Демидова, ул. Советская, 14, Ярославль, 150003, Россия.
Содержание предназначено для детей старше 12 лет.

Editor-in-Chief

Valery A. Sokolov Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)

Deputies Editor-in-Chief

Sergey D. Glyzin Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)

Eugeniy A. Timofeev Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)

Editorial Board Secretary

Egor V. Kuzmin Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)

The Editorial Board

Sergei M. Abramov Professor, Doctor of Sciences, Corresponding Member of Russian Academy of Sciences, Program Systems Institute of RAS (Pereslavl-Zalesskiy, Russia)

Lilian Aveneau Professor, XLIM Laboratory, University of Poitiers (Poitiers, France)

Thomas Baar Professor, Doctor, Hochschule für Technik und Wirtschaft Berlin, University of Applied Sciences (Berlin, Germany)

Olga L. Bandman Professor, Doctor of Sciences, Supercomputer Software Department, Institute of Computational Mathematics and Mathematical Geophysics SB RAS (Novosibirsk, Russia)

Vladimir N. Belykh Professor, Doctor of Sciences, Volga State Academy of Water Transport (Nizhny Novgorod, Russia)

Vladimir A. Bondarenko Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)

Richard R. Brooks Professor, Clemson University (South Carolina, USA)

Alex Dekhtyar Professor, California Polytechnic State University (Cal Poly, California, USA)

Mikhail Dmitriev Professor, Doctor of Sciences, Higher School of Economics (Moscow, Russia)

Vladimir L. Dolnikov Doctor of Sciences, Moscow Institute of Physics and Technology (Moscow, Russia)

Valery G. Durnev Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)

Yuri G. Karpov Professor, Doctor of Sciences, St-Petersburg State Polytechnical University (Russia)

Sergey A. Kashchenko Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)

Lev S. Kazarin Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)

Andrei Yu. Kolesov Professor, Doctor of Sciences, P.G. Demidov Yaroslavl State University (Russia)

Nikolai A. Kudryashov Professor, Doctor of Sciences, MEPhI (Russia)

Olga Kouchnarenko Professor at the Burgundy-Franche-Comte University, The FEMTO-ST Institute (CNRS 6174) (Besancon, France)

Irina A. Lomazova Professor, Doctor of Sciences, Higher School of Economics (Moscow, Russia)

George G. Malinetskiy Professor, Doctor of Sciences, M.V. Keldysh Institute of Applied Mathematics RAS (Moscow, Russia)

Victor E. Malyshkin Professor, Doctor of Sciences, Institute of Computational Mathematics and Mathematical Geophysics SB RAS (Novosibirsk, Russia)

Alexander V. Mikhailov Professor, Doctor of Sciences, University of Leeds, School of Mathematics (Leeds, Great Britain)

Valery A. Nepomniaschy PhD, A.P. Ershov Institute of Informatics Systems SB RAS (Novosibirsk, Russia)

Nikolai Kh. Rozov Professor, Doctor of Sciences, Lomonosov Moscow State University (Russia)

Philippe Schnoebelen Senior Researcher, LSV, CNRS & ENS de Cachan (CACHAN, France)

Natalia Sidorova Dr., Assistant Professor, Architecture of Information Systems group, Technische Universiteit Eindhoven (Eindhoven, Netherlands)

Ruslan L. Smeliansky Professor, Doctor of Sciences, Corresponding Member of RAS, Lomonosov Moscow State University (Russia)

Javid Taheri Associate Professor, Ph.D., Karlstad University (Sweden)

Mark Trakhtenbrot Dr., Holon Institute of Technology (Holon, Israel)

Dimitry Turaev Professor of Applied Mathematics & Mathematical Physics, Imperial College (London, Great Britain)

Vladimir Zakharov Doctor of Sciences, Professor, Lomonosov Moscow State University (Russia)

Главный редактор

В.А. Соколов..... д-р физ.-мат. наук, проф., ЯрГУ (Россия)

Заместители главного редактора

С.Д. Глызин..... д-р физ.-мат. наук, проф., ЯрГУ (Россия)

Е.А. Тимофеев..... д-р физ.-мат. наук, проф., ЯрГУ (Россия)

Ответственный секретарь

Е.В. Кузьмин..... д-р физ.-мат. наук, проф., ЯрГУ (Россия)

Редакционная коллегия

С.М. Абрамов..... д-р физ.-мат. наук, чл.-корр. РАН, Институт программных систем РАН им. А.К. Айламазяна (Россия)

L. Aveneau..... проф., Университет Пуатье (Франция)

T. Baar..... д-р наук, проф., Университет прикладных технических и экономических наук Берлина (Германия)

О.Л. Бандман..... д-р техн. наук, Институт вычислительной математики и математической геофизики СО РАН (Россия)

В.Н. Белых..... д-р физ.-мат. наук, проф., Волжская государственная академия водного транспорта (Россия)

В.А. Бондаренко..... д-р физ.-мат. наук, проф., ЯрГУ (Россия)

R. Brooks..... проф., Университет Клемсона (США)

A. Dekhtyar..... проф., Калифорнийский политехнический университет, департамент компьютерных наук (США)

М.Г. Дмитриев..... д-р физ.-мат. наук, проф., ВШЭ (Россия)

В.Л. Дольников..... д-р физ.-мат. наук, проф., МФТИ (Россия)

В.Г. Дурнев..... д-р физ.-мат. наук, проф., ЯрГУ (Россия)

В.А. Захаров..... д-р физ.-мат. наук, проф., МГУ (Россия)

Л.С. Казарин..... д-р физ.-мат. наук, проф., ЯрГУ (Россия)

Ю.Г. Карпов..... д-р техн. наук, проф., Санкт-Петербургский государственный технический университет (Россия)

С.А. Кашенко..... д-р физ.-мат. наук, проф., ЯрГУ (Россия)

А.Ю. Колесов..... д-р физ.-мат. наук, проф., ЯрГУ (Россия)

Н.А. Кудряшов..... д-р физ.-мат. наук, проф., Засл. деятель науки РФ, МИФИ (Россия)

O. Kouchnarenko..... проф., Университет Бургундии - Франш-Комтэ (Франция)

И.А. Ломазова..... д-р физ.-мат. наук, проф., ВШЭ (Россия)

Г.Г. Малинецкий..... д-р физ.-мат. наук, проф., Институт прикладной математики им. М.В. Келдыша РАН (Россия)

В.Э. Малышкин..... д-р техн. наук, проф., Институт вычислительной математики и математической геофизики СО РАН (Россия)

A. Mikhailov..... д-р физ.-мат. наук, проф., Университет Лидса (Великобритания)

В.А. Непомнящий..... канд. физ.-мат. наук, Институт систем информатики им. А.П. Ершова СО РАН (Россия)

Н.Х. Розов..... д-р физ.-мат. наук, проф., чл.-корр. РАО, МГУ (Россия)

N. Sidorova..... д-р наук, Технический университет Эйндховена (Нидерланды)

P.Л. Смелянский..... д-р физ.-мат. наук, проф., член-корр. РАН, академик РАЕН, МГУ (Россия)

J. Taheri..... доцент, Университет Карлстада (Швеция)

M. Trakhtenbrot..... д-р комп. наук, Холонский технологический институт (Израиль)

D. Turaev..... проф., Имперский колледж Лондона (Великобритания)

Ph. Schnoebelen..... проф., Национальный центр научных исследований и Высшая нормальная школа Кашана (Франция)

Contents

Theory of Computing

<i>Zakharov V. A.</i> Efficient Equivalence Checking Technique for Some Classes of Finite-State Machines	260
--	-----

Theory of Computing

<i>Sokolov V. A.</i> On the Existence Problem of Finite Bases of Identities in the Algebras of Recursive Functions	304
--	-----

Computing Methodologies and Applications

<i>Kuzmin E. V., Gorbunov O. E., Plotnikov P. O., Tyukin V. A., Bashkin V. A.</i> Application of Convolutional Neural Networks for Recognizing Long Structural Elements of Rails in Eddy-Current Defectograms	316
---	-----

Theory of Data

<i>Lagutina K. V., Manakhova A. M.</i> Automated Search and Analysis of the Stylometric Features that Describe the Style of the Prose 19th–21st Centuries.....	330
--	-----

Computer System Organization

<i>Glyzin S. D., Kashchenko S. A., Tolbey A. O.</i> Features of the Algorithmic Implementation of Difference Analogues of the Logistic Equation with Delay.....	344
---	-----

Discrete Mathematics in Relation to Computer Science

<i>Zykin S. V.</i> A Generalization of the Inference Rules for Join Dependencies in Databases	356
---	-----

Содержание

Theory of Computing

<i>Захаров В. А.</i> Эффективные алгоритмы проверки эквивалентности для некоторых классов автоматов.....	260
--	-----

Theory of Computing

<i>Соколов В. А.</i> О проблеме существования конечных базисов тождеств в алгебрах рекурсивных функций.....	304
---	-----

Computing Methodologies and Applications

<i>Кузьмин Е. В., Горбунов О. Е., Плотников П. О., Тюкин В. А., Башкин В. А.</i> Применение свёрточных нейронных сетей для распознавания длинных конструктивных элементов рельсов на вихретоковых дефектограммах.....	316
---	-----

Theory of Data

<i>Лагутина К. В., Манахова А. М.</i> Автоматизированный поиск и анализ стилометрических характеристик, описывающих стиль прозы 19–21 веков.....	330
--	-----

Computer System Organization

<i>Глызин С. Д., Кащенко С. А., Толбей А. О.</i> Особенности алгоритмической реализации разностных аналогов логистического уравнения с запаздыванием.....	344
---	-----

Discrete Mathematics in Relation to Computer Science

<i>Зыкин С. В.</i> Обобщение правил вывода для зависимостей соединения в базах данных.....	356
--	-----

Efficient Equivalence Checking Technique for Some Classes of Finite-State Machines

V. A. Zakharov¹

DOI: [10.18255/1818-1015-2020-3-260-303](https://doi.org/10.18255/1818-1015-2020-3-260-303)

¹National Research University Higher School of Economics (HSE), 20 Myasnitskaya ulitsa, Moscow, 101000, Russia.

MSC2020: 68Q70

Research article

Full text in Russian

Received August 25, 2020

After revision September 7, 2020

Accepted September 9, 2020

Finite transducers, two-tape automata, and biautomata are related computational models descended from the concept of Finite-State Automaton. In these models an automaton controls two heads that read or write symbols on the tapes in the one-way mode. The computations of these three types of automata show many common features, and it is surprising that the methods for analyzing the behavior of automata developed for one of these models do not find suitable utilization in other models. The goal of this paper is to develop a uniform technique for building polynomial-time equivalence checking algorithms for some classes of automata (finite transducers, two-tape automata, biautomata, single-state pushdown automata) which exhibit certain features of the deterministic or unambiguous behavior. This new technique reduces the equivalence checking of automata to solvability checking of certain systems of equations over the semirings of languages or transductions. It turns out that such a checking can be performed by the variable elimination technique which relies on some combinatorial and algebraic properties of prefix-free regular languages. The main results obtained in this paper are as follows:

1. Using the algebraic approach a new algorithm for checking the equivalence of states of deterministic finite automata is constructed; time complexity of this algorithm is $O(n \log n)$.
2. A new class of prefix-free finite transducers is distinguished and it is shown that the developed algebraic approach provides the equivalence checking of transducers from this class in quadratic time (for real-time prefix-free transducers) and cubic (for prefix-free transducers with ϵ -transitions) relative to the sizes of analysed machines.
3. It is shown that the equivalence problem for deterministic two-tape finite automata can be reduced to the same problem for prefix-free finite transducers and solved in cubic time relative to the size of the analysed machines.
4. In the same way it is proved that the equivalence problem for deterministic finite biautomata can be solved in cubic time relative to the sizes of analysed machines.
5. By means of the developed approach an efficient equivalence checking algorithm for the class of simple grammars corresponding to deterministic single-state pushdown automata is constructed.

Keywords: transducer; two-tape automaton; biautomaton; simple grammar; equivalence checking; prefix-free language; language equation; decision procedure.

INFORMATION ABOUT THE AUTHORS

Vladimir Anatolyevich Zakharov | orcid.org/0000-0002-3794-9565. E-mail: zakh@cs.msu.ru
correspondence author | Doctor of Science, Professor.

Funding: This work was supported by the Russian Foundation for Basic Research, Grant N 18-01-00854.

For citation: V. A. Zakharov, "Efficient Equivalence Checking Technique for Some Classes of Finite-State Machines", *Modeling and analysis of information systems*, vol. 27, no. 3, pp. 260-303, 2020.

Эффективные алгоритмы проверки эквивалентности для некоторых классов автоматов

В. А. Захаров¹

DOI: [10.18255/1818-1015-2020-3-260-303](https://doi.org/10.18255/1818-1015-2020-3-260-303)

¹Национальный исследовательский университет «Высшая школа экономики», ул. Мясницкая, 20, г. Москва, 101000 Россия.

УДК 517.9

Научная статья

Полный текст на русском языке

Получена 25 августа 2020 г.

После доработки 7 сентября 2020 г.

Принята к публикации 9 сентября 2020 г.

Конечные преобразователи, двухленточные автоматы и биавтоматы — взаимосвязанные вычислительные модели, ведущие свое происхождение от концепции конечного автомата. В вычислениях этих машин проявляется много общих черт, и удивительно, что методы анализа, разработанные для одной из указанных моделей, не находят подходящего применения в других моделях. Целью данной статьи является разработка единой методики построения быстрых алгоритмов проверки эквивалентности для некоторых классов автоматов (конечных преобразователей, двухленточных автоматов, биавтоматов, магазинных автоматов), которые демонстрируют определенные черты детерминированного или однозначного поведения. Этот новый метод сводит проверку эквивалентности автоматов к проверке разрешимости систем уравнений над полукольцами языков или бинарных отношений. Как оказалось, такую проверку достаточно просто провести методом исключения переменных, используя некоторые комбинаторные и алгебраические свойства регулярных префиксных языков. Основные результаты, полученные в этой статье, таковы.

1. При помощи алгебраического метода построен новый алгоритм проверки эквивалентности детерминированных конечных автоматов, имеющий сложность по времени $O(n \log n)$.
2. Выделен новый класс префиксных конечных трансдюсеров и показано, что проверка эквивалентности трансдюсеров из этого класса может быть осуществлена новым методом за время, квадратичное (для префиксных трансдюсеров реального времени) и кубическое (для префиксных трансдюсеров с ε -переходами) относительно размеров анализируемых автоматов.
3. Показано, что проблема эквивалентности для детерминированных двухленточных конечных автоматов сводится к задаче проверки эквивалентности префиксных конечных трансдюсеров и может быть решена за время, кубическое относительно их размеров.
4. Аналогичным образом установлена разрешимость проблемы эквивалентности для детерминированных конечных биавтоматов за время, кубическое относительно их размеров.
5. При помощи нового метода построен алгоритм проверки эквивалентности для простых грамматик, соответствующих детерминированным магазинным автоматам с одним состоянием.

Ключевые слова: трансдюсер; двухленточный автомат; биавтомат; простая грамматика; проверка эквивалентности; префиксный язык; языковое уравнение; разрешающая процедура.

ИНФОРМАЦИЯ ОБ АВТОРАХ

Владимир Анатольевич Захаров	orcid.org/0000-0002-3794-9565 . E-mail: zakh@cs.msu.ru
автор для корреспонденции	Доктор физ.-мат. наук, профессор.

Финансирование: Работа выполнена при финансовой поддержке гранта РФФИ N 18-01-00854.

Для цитирования: V. A. Zakharov, "Efficient Equivalence Checking Technique for Some Classes of Finite-State Machines", *Modeling and analysis of information systems*, vol. 27, no. 3, pp. 260-303, 2020.

Введение

Конечные автоматы-преобразователи (трансдюсеры), двухленточные автоматы с конечным числом состояний (2-FSAs) и конечные биавтоматы — это взаимосвязанные модели вычислений, ведущие свое происхождение от концепции конечного автомата. В этих моделях программа с конечным числом состояний управляет двумя головками, каждая из которых считывает или записывает буквы на лентах, перемещаясь при этом в одну сторону. Конечные трансдюсеры и двухленточные автоматы вычисляют один и тот же класс рациональных бинарных отношений на словах, но вычисления в этих моделях проводятся по-разному: трансдюсер считывает слова на входной ленте и делает записи на выходной ленте, а 2-FSA считывает слова попеременно на двух входных лентах. Обе модели допускают взаимную трансляцию друг в друга, и поэтому многие авторы не делают различия между ними, когда речь заходит о рациональных отношениях. Биавтоматы проводят вычисления на единственной входной ленте и способны распознавать линейные контекстно-свободные языки, прочитывая входные слова сразу с двух концов. Взаимосвязь между конечными трансдюсерами и биавтоматами основана на простом факте, отмеченном в статье [1]: для каждого рационального бинарного отношения R множество слов $L = \{uv^{-1} : (u, v) \in R\}$ является линейным контекстно-свободным языком.

Указанные модели, несмотря на сходство их вычислительных возможностей, имеют разные области применения. При изучении и применении автоматов-преобразователей обычно ограничиваются рассмотрением трансдюсеров реального времени; эти автоматы проводят запись на выходной ленте только в ответ на считывание очередного символа на входной ленте. Трансдюсеры реального времени находят применение во многих областях информатики, таких как: вычислительная лингвистика и обработка текстов [2], биоинформатика [3], проектирование реагирующих и распределенных систем [4–7]. Задачи оптимизации и верификации информационных систем и математических моделей, возникающие в этих приложениях, могут быть сведены к проблемам проверки эквивалентности и минимизации конечных автоматов-преобразователей (см. [7–9]).

Область применения многоленточных автоматов (m-FSAs) не столь обширна. В статье [10] было показано, что задача оптимизации строковых предикатов, которые используются для представления и обработки строк в базах данных, может быть сведена к проблеме минимизации для m-FSAs. В статье [11] было отмечено, что проблема проверки эквивалентности для стандартных схем программ, содержащих только унарные предикаты вида $p(x)$ и операторы присваивания вида $x := f(x)$, взаимосводима к проблеме эквивалентности для детерминированных многоленточных автоматов (m-DFSAs). Трудность решения последней из указанных задач оказала значительное влияние на развитие математических методов оптимизации программ.

Биавтоматы были впервые введены в статье [12] и независимо в работе [13] (в первых публикациях они были названы линейными автоматами) как более простая альтернатива магазинным автоматам (PDAs) для обработки контекстно-свободных (к-с) языков, порожденных линейными грамматиками. Впоследствии в нескольких публикациях было проведено подробное изучение вычислительных возможностей биавтоматов [14–16] и алгоритмических аспектов задач анализа их поведения [14, 17]. Биавтоматы относятся к числу наиболее простых моделей вычислений, способных распознавать нерегулярные к-с языки, и можно полагать, что они найдут подходящее применение в решении задач синтаксического анализа и обработки текстов. В этом случае будет весьма полезно иметь эффективные алгоритмы проверки эквивалентности и минимизации для биавтоматов.

Упомянутые здесь три модели вычислений имеют немало общих черт, и поэтому вызывает удивление то обстоятельство, что алгоритмические задачи анализа их поведения до сих пор не рассматривались с единой точки зрения. Главная цель этой статьи — представить вниманию читателей общий метод построения эффективных (в том числе полиномиальных по времени) алгоритмов про-

верки эквивалентности для нескольких взаимосвязанных классов автоматов (конечных трансдюсеров, двухленточных автоматов, биавтоматов, магазинных автоматов с единственным состоянием управления), вычисления которых обладают свойствами детерминированности или однозначности. Чтобы повернуть Землю, нужна точка опоры. В поисках ее целесообразно провести обзор известных ранее результатов, касающихся разрешимости проблемы эквивалентности для конечных трансдюсеров, многоленточных и магазинных автоматов, и попытаться разработать такой подход, который был бы пригоден для всех этих автоматных моделей. Вначале мы рассмотрим ранее известные результаты исследования задачи проверки эквивалентности конечных автоматов-преобразователей, поскольку изучению и решению этой проблемы для разнообразных классов трансдюсеров было посвящено довольно много работ.

Исследование проблемы эквивалентности для трансдюсеров (машин с конечным множеством состояний) проводится с начала 60-х. Его можно условно разделить на три периода. Вначале П. Фишер и А. Розенберг [18] доказали алгоритмическую неразрешимость этой задачи для конечных трансдюсеров произвольного вида путем сведения к ней проблемы соответствий Поста. Вслед за этим Т. Гриффитс [19] получил аналогичный результат для класса трансдюсеров реального времени, и немного позднее О. Ибарра [20] сумел показать, что алгоритмическая неразрешимость проблемы эквивалентности сохраняется даже в том случае, когда трансдюсеры работают над однобуквенным выходным алфавитом.

Доказательства, приведенные в статьях [18–20], показывают, что неразрешимость возникает в тех случаях, когда некоторые входные слова могут иметь произвольно много образов. Поэтому на следующем этапе исследований внимание было сосредоточено на изучении проблемы эквивалентности для трансдюсеров с ограниченным числом значений. Вначале удалось установить, что эта проблема разрешима для функциональных [21, 22], детерминированных [23] и конечно неоднозначных [24] трансдюсеров. Кроме того, в статье [24] было показано, что проверку эквивалентности можно провести за полиномиальное время, если анализируемые трансдюсеры являются однозначными и, в частности, детерминированными. Были также предложены полиномиальные по времени алгоритмы проверки того, является ли заданный трансдюсер реального времени ограниченно-значным [25], или k -значным для любого фиксированного k [24]. Трансдюсер считается k -значным, если для любого входного слова множество его образов содержит не более k выходных слов. Что касается проблемы эквивалентности для k -значных трансдюсеров, ее разрешимость была установлена в статье [26], однако предложенная в этой работе разрешающая процедура не позволяла получить оценки сложности решения этой проблемы. Более совершенный подход, основанный на представлении произвольного k -значного трансдюсера в виде прямой суммы конечного числа функциональных трансдюсеров, был предложен А. Вебером в статье [27] (см. также [28]). Ему удалось получить верхнюю оценку длины входного слова, на котором может проявляться неэквивалентность заданных k -значных трансдюсеров и, таким образом, показать, что проблема эквивалентности для этого класса трансдюсеров разрешима за время, ограниченное двойной экспонентой от размера анализируемых автоматов. Разрешающая процедура, описанная в статье [28], основывается исключительно на комбинаторных соображениях, и этим объясняется столь высокая верхняя оценка ее сложности.

В стремлении уменьшить сложность процедур проверки эквивалентности для ограниченно-значных трансдюсеров Ж. Сакарович и Р. де Суза провели систематическую ревизию методов, предложенных в работах [27, 28]. В серии статей [29–33] им удалось модифицировать метод декомпозиции k -значных трансдюсеров и разработать улучшенный способ проверки эквивалентности, который в значительно большей мере принимает в расчет структуру анализируемых трансдюсеров. Предложенные усовершенствования позволили построить полиномиальные по времени процедуры проверки свойства k -значности трансдюсеров реального времени и экспоненциальные

по времени разрешающие процедуры для проверки эквивалентности k -значных трансдюсеров. Существенно более простой метод верификации был описан в статье [34]. Для его применения не требуется проводить предварительную декомпозицию рассматриваемых трансдюсеров, он работает непосредственно с их отношениями переходов и позволяет анализировать поведение трансдюсеров реального времени, которые оперируют не только над выходными словами, но и над полугруппами. В работе [34] было показано, что эквивалентность k -значных трансдюсеров можно проверить за экспоненциальное время в том случае, если полугруппа выходных элементов вложима в группу, в которой проблема проверки тождеств разрешима за полиномиальное время.

Как видно из приведенного здесь краткого обзора по проблеме эквивалентности для конечных автоматов-преобразователей, ограниченно-значные трансдюсеры долгое время находились в центре внимания. Главная цель исследований состояла в проведении как можно более точной демаркации границы между разрешимыми и неразрешимыми случаями проблемы эквивалентности. Оказалось, что наиболее обширным среди известных классов преобразователей, для которых проблема эквивалентности разрешима, является класс преобразователей с ограниченной степенью длины (см. [35]). В статье [36] было показано, что эквивалентность трансдюсеров из этого класса можно проверить за время, которое оценивается сверху тройной экспонентой, зависящей от размера анализируемых автоматов. Столь высокая трудоемкость разрешающей процедуры вряд ли позволит использовать ее в каких-либо приложениях. Возможно, разумнее было бы изменить цель исследования и попытаться обнаружить классы трансдюсеров, для которых проблема эквивалентности допускает решение за практически приемлемое время. В статье [24] было показано, что эквивалентность однозначных трансдюсеров может быть проверена за полиномиальное время. Однако авторы статьи [24] рассматривали свойство однозначности (unambiguity) конечных трансдюсеров лишь применительно к действиям считывания символов на входной ленте. Возможно, что ослабление этого свойства позволит выделить новый, более обширный класс автоматов, для которого проблема эквивалентности будет также разрешима за полиномиальное время.

Неразрешимость проблемы эквивалентности для недетерминированных 2-FSAs была доказана в статье [18]. Но вскоре М. Берд [37] показал, что эта задача разрешима для 2-DFSAs. Аналогичный вывод следует также из результатов работы Л. Вэлианта [38]. Интересная особенность 2-DFSAs состоит в том, что проблема включения для этого класса автоматов неразрешима (см. [37]). Автор статьи [39] улучшил алгоритм Вэлианта и получил верхнюю оценку временной сложности $2^{O(n^6)}$. Позднее Э. Фридман и Ш. Грейбах обнаружили, что эквивалентность 2-DFSAs можно проверять за полиномиальное время. В статье [40] они предложили два алгоритма. Первый из них прост для понимания, но он имеет экспоненциальную временную сложность. Более изощренный второй алгоритм основан на идеях первого и проверяет эквивалентность 2-DFSA за время $O(n^{12})$. Авторы статьи [40] также коротко пояснили, как можно улучшить этот алгоритм, чтобы достичь временной сложности $O(n^4)$. И, наконец, в 1991 г. разрешимость проблемы эквивалентности для детерминированных автоматов с произвольным числом лент была установлена Т. Харью и Ю. Кархюмяки в статье [41]. Используя серию замечательных результатов из теории групп, они показали, как проверить, имеют ли два недетерминированных многоленточных автомата одинаковое количество допускающих прогонов на каждом входе. Из описания разрешающей процедуры, предложенной в [41], видно, что проблема эквивалентности для n -DFSA принадлежит классу сложности **co-NP**. Как ни странно, но за прошедшие три десятилетия в изучении этой проблемы не было достигнуто значительного прогресса. Тем не менее, необходимо отметить статью [42]; ее автор переформулировал стратегию принятия решений, разработанную в работе [41], в терминах матричной алгебры и построил рандомизированный алгоритм для проверки эквивалентности n -DFSA за полиномиальное время.

Возможно, дальнейший прогресс в изучении n -DFSAs может быть достигнут путем установления и использования отношений между многоленточными автоматами и другими вычислительными моделями. Учитывая тесную взаимосвязь между конечными преобразователями и 2-FSAs, можно предположить, что процедуры проверки эквивалентности, разработанные для определенных классов конечных преобразователей, могут быть адаптированы для анализа 2-DFSAs.

Вычислительные возможности биавтоматов изучались в работах [12, 14, 43]. Р. Луканова [12] доказала, что недетерминированные биавтоматы могут распознавать все k -с языки, порождаемые линейными грамматиками. Что касается вычислительных возможностей детерминированных биавтоматов, в статье [43] было показано, что эта модель вычислений несравнима с детерминированными магазинными автоматами (DPDAs): при помощи детерминированных биавтоматов можно распознавать язык палиндромов, который не принадлежит семейству детерминированных k -с языков, хотя детерминированные биавтоматы не могут распознавать нелинейные k -с языки. Авторы статьи [14] провели подробное исследование детерминированных биавтоматов и получили много интересных алгоритмических и теоретико-множественных результатов о свойствах этой модели вычислений. Одна из наиболее значимых задач, которая оставалась нерешенной (см. [14]), — это проблема эквивалентности для детерминированных биавтоматов. И хотя глубокое изучение этой задачи фактически еще не начиналось, интенсивные исследования аналогичной проблемы были проведены для некоторых других родственных вычислительных моделей, таких как магазинные автоматы и k -с грамматики.

В статье [44] было доказано, что проблема эквивалентности для k -с грамматик и магазинных автоматов алгоритмически неразрешима. Из доказательства видно, что неразрешимость имеет место для линейных k -с грамматик (а, значит, и для недетерминированных биавтоматов). Но спустя несколько лет А. Кореньяк и Дж. Хопкрофт [45] обнаружили, что проблема эквивалентности разрешима для $LL(1)$ -грамматик в нормальной форме Грейбах (т.н. простых грамматик), которые соответствуют детерминированным магазинным автоматам с единственным состоянием управления. Разрешающая процедура, предложенная в статье [45], проверяет разрешимость систем языковых уравнений, соответствующих правилам вывода анализируемых грамматик. Было показано, что $LL(1)$ -языки обладают свойством префиксности, и оно позволяет проводить проверку разрешимости указанных систем уравнений с помощью равносильных преобразований. Авторы статьи [45] показали, что эти равносильные преобразования можно применять так, чтобы размер систем уравнений экспоненциально зависел от размера анализируемых грамматик; таким образом, время работы разрешающего алгоритма оценивается сверху двойной экспонентой от размера проверяемых грамматик. Стоит отметить также, что, как и в случае многоленточных автоматов, проблема включения для простых грамматик алгоритмически неразрешима (см. [46]). Д. Кокаль в статье [47] сумел преодолеть эффект двойного экспоненциального взрыва. Он следовал тем же путем, что и авторы работы [45], но использовал более изощренные свойства префиксных языков в равносильных преобразованиях систем языковых уравнений. Кокаль показал, как можно проверять разрешимость таких систем уравнений с помощью метода исключения переменных, и сократил временную сложность разрешающей процедуры до одинарной экспоненты. Присутствие экспоненциального множителя в этой оценке обусловлено тем, что длина кратчайшего слова, которое порождается k -с грамматикой и определяет размер системы уравнений, может зависеть экспоненциально от размера грамматики. Исследовательская группа в составе С. Бастьена, Ю. Чижовича и др. в статье [48] сумела уменьшить оценку сложности алгоритма Кокаля, воспользовавшись полиномиальным алгоритмом сравнения пар слов, представленных в сжатой форме т.н. “прямыми программами” (k -с грамматиками, порождающими языки, которые состоят из одного слова), и получили, таким образом, полиномиальный по времени алгоритм проверки эквивалентности простых грамматик (ранее полиномиальный алгоритм более высокой сложности был предложен в статье [49]).

Приведенный обзор позволяет сделать несколько важных выводов.

За редким исключением, все работы, посвященные исследованию проблемы эквивалентности для указанных выше автоматных моделей вычислений, имеют фрагментарный характер: авторы выделяют некоторый класс автоматов и пытаются построить алгоритм проверки эквивалентности, существенно использующий особенности вычислений автоматов этого класса. Несмотря на хорошо известные и вполне очевидные взаимосвязи вычислений автоматов из разных классов, очень редко предпринимались попытки перенести результаты и адаптировать методы решения проблемы эквивалентности, полученные для одних классов автоматов, на другие классы моделей вычислений. В связи с этим можно задаться вопросом о возможности создания такого метода проверки эквивалентности автоматных моделей вычислений, который, с одной стороны, обладал бы определенной универсальностью и был бы в равной мере применим к автоматам разных типов, а, с другой стороны, позволял бы получать эффективные алгоритмы решения проблемы эквивалентности за счет учета специфических особенностей вычислений некоторых классов автоматов.

В большинстве работ авторы преследовали вполне естественную цель выделить как можно более обширные классы автоматов с разрешимой проблемой эквивалентности. Однако по мере расширения “области разрешимости” этой проблемы алгоритмы проверки эквивалентности автоматов становились все более трудоемкими и менее практичными. Подходящим примером здесь могут служить магазинные автоматы. Разрешимость проблемы эквивалентности для сравнительно узких классов DPDAs была установлена в статьях [38, 45, 50]; предложенные алгоритмы имели не менее чем экспоненциальную временную сложность. Впоследствии для этих классов автоматов были построены полиномиальные по времени или имеющие сложность, близкую к этой оценке, алгоритмы проверки эквивалентности (см. [48, 51, 52]). В то же время, многие исследователи [53–59] неустанно расширяли “область разрешимости” проблемы эквивалентности для DPDAs. Методы построения разрешающих алгоритмов становились все более изощренными и, наконец, Ж. Синезерг [60] сумел доказать разрешимость проблемы эквивалентности для произвольных DPDAs. Однако предложенная разрешающая процедура оказалась чрезвычайно трудоемкой и совершенно непрактичной: единственное, что удалось выяснить о ее сложности — это то, что она может быть оценена примитивно рекурсивной функцией [61]. Этот пример показывает, что нередко по мере расширения области применения разрешающих алгоритмов их прикладная ценность значительно снижается. Таким образом, желательно разработать такой метод построения алгоритмов проверки эквивалентности различных типов автоматов, который позволял бы гарантировать высокую эффективность разрешающих процедур, построенных с его помощью, при выполнении определенных требований, которые налагаются на устройство анализируемых автоматов. Как можно видеть из статьи [62], такой подход может приводить к довольно интересным практическим результатам.

Для достижения поставленной цели нужно выбрать оптимальный подход к построению разрешающих алгоритмов. Среди разнообразия методов проверки эквивалентности программ можно выделить три подхода, имеющих наиболее широкое применение: метод оценки длины контрпримера, метод анализа совместных вычислений и метод проверки разрешимости систем уравнений. В первом из них оценивается длина кратчайшего контрпримера — входных данных, на которых неэквивалентные программы заданного размера или структуры вычисляют разные результаты. Если такая оценка известна, то для проверки эквивалентности двух программ (автоматов) достаточно сравнить результаты их вычислений на всех входных данных, размер которых не превосходит этой оценки. Этот метод был успешно использован в работах [21, 27, 28, 41]. Недостаток метода состоит в том, что он требует переборной проверки большого объема входных данных, и с его помощью нельзя построить быстрых алгоритмов верификации. В методе совместных вычислений для пары проверяемых программ (автоматов) строится система переходов, вычисления которой моделируют

всевозможные пары вычислений этих программ на одинаковых входных данных. В построенной системе переходов допускающим объявляется такое состояние совместных вычислений пары программ, достижение которого явно свидетельствует о том, что эти программы способны получать на одинаковых данных разные результаты. Таким образом, задача проверки эквивалентности программ сводится к задаче проверки пустоты в том классе систем переходов, которые моделируют совместные вычисления этих программ.

Метод совместных вычислений нашел применение во многих работах [33, 34, 36–38, 40, 51, 55, 57, 59]. Наибольшую трудность в его использовании составляет выбор подходящей системы переходов для моделирования совместных вычислений двух программ: такие системы должны быть достаточно “подробными”, чтобы соответствовать семантике проверяемых программ и различать результаты их вычислений, и, вместе с тем, они должны быть достаточно простыми для эффективного решения проблемы пустоты. Например, совместные вычисления двух DPDAs можно очень просто моделировать при помощи автоматов с двумя магазинами, но проблема пустоты для этой модели вычислений неразрешима. Одна из заслуг Л. Вэлианта состоит в том, что в статье [38] ему удалось выделить такой класс DPDAs, совместные вычисления которых можно моделировать при помощи автоматов с одним магазином, и, таким образом, получить решение проблемы эквивалентности для DPDAs с конечным числом поворотов магазина. Изучению необходимых и достаточных условий, которым должны удовлетворять системы переходов для успешного их применения в методе совместных вычислений, посвящена статья [63].

В алгебраическом методе проверки эквивалентности вычисления любой программы (автомата) описываются при помощи системы уравнений: каждой точке (состоянию управления) программы ставится в соответствие переменная, и уравнения системы описывают зависимость результатов, вычисляемых программой в одних состояниях, от данных, которые были вычислены программой в смежных состояниях. Если к такой системе добавить уравнение, приравнивающее друг другу переменные, соответствующие выходам этих программ, то получим систему уравнений, разрешимость которой равносильна эквивалентности анализируемых программ. Поэтому для проверки эквивалентности программ достаточно уметь проверять разрешимость соответствующих систем уравнений. Преимущества этого подхода ясны: алгебраизация задачи позволяет воспользоваться для ее решения обширным арсеналом выразительных средств и методов современной алгебры. Алгебраический метод проверки эквивалентности использовался в статьях [45, 47, 48, 52, 58, 60, 64]. Взаимосвязь проблем эквивалентности для автоматов и разрешимости уравнений над множествами слов была исследована в статье [65].

Произвольные системы языковых уравнений трудны для решения — с их помощью можно задавать любые рекурсивно перечислимые языки. Но, как показано в статье [66], даже системы линейных языковых уравнений в общем случае эффективно неразрешимы. Существенную роль здесь играют свойства языков, используемых в уравнениях в качестве коэффициентов. Одним из таких полезных свойств является свойство префиксности языков — язык L называется префиксным, если ни одно слово w из L не является префиксом других слов этого языка. Опираясь именно на свойство префиксности языков, авторам статьи [45] удалось доказать разрешимость проблемы эквивалентности для простых k -с языков. В последующих работах [47, 48], посвященных изучению этой задачи, префиксное свойство также сыграло решающую роль. Важное значение этого свойства для эффективного решения языковых уравнений отмечалось в работе [65]: была доказана разрешимость проблемы эквивалентности конечных префиксных подстановок на заданных k -с языках, а также проблемы эквивалентности для одного класса недетерминированных трансдюсеров. При помощи префиксного свойства Ж. Сенизергу [52] удалось показать, что проблема эквивалентности для DPDA с конечным числом поворотов магазина принадлежит классу сложности **co-NP**. Перечисленные результаты свидетельствуют о том, что для построения эффективных алгоритмов проверки

эквивалентности целесообразно обратить внимание на классы автоматов, устройство которых тесно связано с префиксными языками.

Цель данной статьи — выделить такие классы автоматов (трансдюсеров, многоленточных автоматов, биавтоматов, магазинных автоматов), для которых можно получить эффективное решение проблемы эквивалентности при помощи алгебраического метода, опираясь на свойства префиксности регулярных языков.

Основные результаты, полученные в этой статье, таковы.

1. Разработан новый метод проверки эквивалентности автоматов различных типов; в этом методе задача проверки эквивалентности автоматов сводится к задаче проверки разрешимости систем линейных языковых уравнений, и эта задача решается при помощи традиционных алгебраических приемов с существенным использованием свойств регулярных префиксных языков.
2. При помощи алгебраического метода построен новый алгоритм проверки эквивалентности состояний детерминированных конечных автоматов, имеющий сложность по времени $O(n \log n)$.
3. Выделен новый класс префиксных конечных трансдюсеров и показано, что проверка эквивалентности трансдюсеров из этого класса может быть осуществлена новым методом за время, квадратичное (для префиксных трансдюсеров реального времени) и кубическое (для префиксных трансдюсеров общего вида) относительно размеров анализируемых автоматов.
4. Показано, что проблема эквивалентности для детерминированных двухленточных конечных автоматов сводится к задаче проверки эквивалентности префиксных конечных трансдюсеров и может быть решена за время, кубическое относительно их размеров.
5. Установлена разрешимость проблемы эквивалентности для детерминированных конечных биавтоматов за время, кубическое относительно их размеров.

В этой статье также показано, что предложенный метод проверки разрешимости систем языковых уравнений можно также использовать для анализа систем нелинейных уравнений и с его помощью получать простые алгоритмы проверки эквивалентности для некоторых классов магазинных автоматов и k -с грамматик, в частности, $LL(1)$ -грамматик и соответствующих им детерминированных магазинных автоматов с единственным состоянием управления.

Статья организована следующим образом. Раздел 1 содержит минимальный набор основных понятий из теории формальных языков, которые являются общими для всех последующих разделов. В разделе 2 на примере простейшей модели детерминированных конечных автоматов описаны основные положения того алгебраического метода проверки эквивалентности автоматных моделей вычислений, которому посвящена эта статья. Применение описанного метода для более сложных типов автоматов, которые рассматриваются в данной статье, требует использования некоторых свойств регулярных префиксных языков. Поэтому следующий раздел 3 посвящен префиксным языкам: доказаны свойства замкнутости класса префиксных языков относительно некоторых теоретико-языковых операций (конкатенации, итерации, деления на слова слева и справа), рассмотрены автоматные способы описания префиксных языков и установлены оценки сложности выполнения указанных операций над префиксными языками, описанными при помощи конечных автоматов. В разделе 4 введены основные понятия теории конечных автоматов-преобразователей (обобщенных машин с конечным числом состояний, трансдюсеров). В разделе 5 описан новый класс трансдюсеров реального времени — префиксных трансдюсеров — и показано, как проблема эквивалентности для трансдюсеров из этого класса может быть сведена к задаче проверки разрешимости систем линейных языковых уравнений, в которых коэффициентами и свободными членами являются конечные слова, образующие конечные префиксные языки. В этом же разделе описан квадратичный по сложности алгоритм проверки разрешимости указанных систем

языковых уравнений; в его основу положен метод Гаусса исключения переменных, используемый в линейной алгебре для решения систем линейных уравнений. В этом алгоритме существенно используются свойства префиксных языков, которые были установлены в разделе 3. В следующем разделе 6 исследован еще более обширный класс недетерминированных трансдюсеров, в которых допустимы переходы без считывания входных букв — класс префиксных однородных трансдюсеров с ε -переходами. В линейных уравнениях, описывающих поведение трансдюсеров из этого класса, коэффициентами и свободными членами являются регулярные префиксные языки, которые представлены детерминированными конечными автоматами. Процедура проверки разрешимости таких систем уравнений, основанная на методе исключения переменных, несколько усложняется. Она так же использует особенности префиксных языков, отмеченные в разделе 3, и позволяет проверять эквивалентность префиксных однородных трансдюсеров с ε -переходами за время, кубическое относительно их размеров. Далее, в разделе 7 исследована проблема эквивалентности для детерминированных двухленточных автоматов. Показано, что класс бинарных отношений (трансдукций), распознаваемых автоматами из этого класса, совпадает с классом бинарных отношений, вычисляемых префиксными однородными трансдюсерами с ε -переходами. Обнаруженное соответствие позволяет провести трансляцию детерминированных двухленточных автоматов в префиксные однородные трансдюсеры и, воспользовавшись результатами раздела 6, построить кубический по времени алгоритм проверки эквивалентности детерминированных двухленточных автоматов. В разделе 8 метод проверки разрешимости систем линейных языковых уравнений, разработанный в разделе 3, подходящим образом адаптирован для решения проблемы эквивалентности детерминированных конечных биавтоматов; показано, что эта задача может быть решена за время, кубическое относительно размеров проверяемых биавтоматов. Метод исключения переменных применим не только для проверки разрешимости систем линейных языковых уравнений; в разделе 9 показано, как при помощи этого метода можно эффективно проверять разрешимость систем языковых уравнений, порожденных $LL(1)$ -грамматиками. Здесь так же решающую роль играет свойство префиксности $LL(1)$ -языков. Перспективы дальнейшего развития и применения предложенного в этой статье алгебраического подхода к разработке алгоритмов проверки эквивалентности различных типов автоматов обсуждаются в заключительном разделе 9.

1. Основные понятия

Алфавитом называется всякое конечное непустое множество букв $\Sigma = \{a_1, \dots, a_k\}$. *Слово* в алфавите Σ — это любая конечная последовательность $w = a_{i_1} a_{i_2} \dots a_{i_n}$ букв из Σ . Пустое слово обозначим символом ε , а множество всех слов в алфавите Σ обозначим записью Σ^* . Длина $|w|$ слова w — это количество букв в w . Для произвольной пары слов $u = a_{i_1} \dots a_{i_n}$ и $v = a_{j_1} \dots a_{j_m}$ запись uv будет обозначать их *конкатенацию*, которая является словом $a_{i_1} \dots a_{i_n} a_{j_1} \dots a_{j_m}$. Для любого слова $w = a_{i_1} a_{i_2} \dots a_{i_n}$ запись w^{-1} будет обозначать *обратное слово* $a_{i_n} \dots a_{i_2} a_{i_1}$.

Предположим, что слово w является конкатенацией слов u и v , т.е. $w = uv$. Тогда слово u является *префиксом* w , слово v — *суффиксом* w , а слово w — *расширением* слова u . В том случае, если $w = uv$ мы будем называть слово u *частным при делении справа* слова w на слово v (и обозначать записью $u = w/v$), а слово v — *частным при делении слева* слова w на слово u (и обозначать записью $v = w \setminus u$). Два слова w_1 и w_2 будут называться *совместными*, если одно из них является префиксом другого; в противном случае эти слова будут считаться *несовместными*.

Язык L в алфавите Σ — это любое подмножество множества слов Σ^* . *Конкатенацией языков* L_1 и L_2 является язык $L_1 L_2 = \{uv : u \in L_1, v \in L_2\}$. Если $L_1 = \emptyset$ или $L_2 = \emptyset$, то $L_1 L_2 = \emptyset$. Если $L_1 = \{u\}$, то мы будем использовать запись uL_2 для обозначения конкатенации L_1 и L_2 . *Итерацией языка* L будем называть язык $L^* = \{\varepsilon\} \cup L \cup LL \cup \dots$. *Обращением языка* L будем называть язык $L^{-1} = \{w^{-1} : w \in L\}$. *Частным при делении справа* языка L на слово w считается язык $L/w = \{u : uw \in L\}$ частных при делении справа всех слов языка L на слово w , а *частным при делении слева* языка L на слово w

считается язык $L \setminus w = \{u : wu \in L\}$ частных при делении слева всех слов языка L на слово w . Необходимо иметь в виду, что в том случае, если ни одно слово из L не имеет w в качестве суффикса (префикса), то $L/w = \emptyset$ (соответственно, $L \setminus w = \emptyset$).

Трансдукцией в алфавитах Σ и Δ называется всякое подмножество T множества пар слов $\Sigma^* \times \Delta^*$. *Областью определения* трансдукции T называется язык $Dom(T) = \{u : \exists w : (u, w) \in T\}$, а *образом* трансдукции T называется язык $Im(T) = \{w : \exists u : (u, w) \in T\}$. Язык $Im(T, u) = \{w : (u, w) \in T\}$ называется *образом слова u в трансдукции T* . Конкатенацией трансдукций R_1 и R_2 является трансдукция $R_1 R_2 = \{(u_1 u_2, v_1 v_2) : (u_1, v_1) \in R_1, (u_2, v_2) \in R_2\}$.

Обозначим записью $Reg(\Sigma)$ семейство регулярных языков в алфавите Σ . Регулярные языки обычно описываются с помощью конечных автоматов. *Детерминированный конечный автомат* (DFA) в алфавите Σ задается системой переходов $A = \langle Q, q_0, F, \varphi \rangle$, в которой Q — это конечное множество состояний, q_0 — начальное состояние, $F \subseteq Q$ — подмножество допускающих состояний, и $\varphi : Q \times \Sigma \rightarrow Q$ — частичная функция переходов. Тройки (q, x, q') , в которых $\varphi(q, x) = q'$, называются *переходами* DFA A и обозначаются записями $q \xrightarrow{x} q'$. DFA A допускает слово $w = a_1 a_2 \dots a_n$, если существует такая последовательность переходов $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \dots \xrightarrow{a_{n-1}} q_{n-1} \xrightarrow{a_n} q_n$, в которой q_0 является начальным состоянием, а q_n — допускающим состоянием автомата A . DFA A описывает регулярный язык $L(A)$, представляющий собой множество всех слов w , которые допускаются автоматом A . *Размер* DFA A — это число его состояний $|Q|$. Далее DFA A с начальным состоянием q условимся обозначать записью $A(q)$.

Для описания конечных семейств регулярных языков мы будем использовать *детерминированные конечные автоматы с несколькими множествами допускающих состояний* (multi-DFAs) в алфавите Σ . Такой автомат задается системой переходов $D = \langle Q, q_0, F_1, \dots, F_m, \varphi \rangle$, в которой Q , q_0 , и φ имеют тот же смысл, что и для DFAs, а $F_1, \dots, F_m$ — это подмножества допускающих состояний. Такой multi-DFA D описывает семейство регулярных языков $\{L(D_1), \dots, L(D_m)\}$, где $D_i = \langle Q, q_0, F_i, \varphi \rangle$ — это обычный DFA для каждого $i, 1 \leq i \leq m$.

Вполне естественно ожидать, что DFAs и multi-DFAs не имеют бесполезных состояний. DFA (multi-DFA) A называется *сокращенным*, если его допускающие состояния достижимы из каждого состояния автомата A . В дальнейшем по умолчанию будем подразумевать, что все DFAs и multi-DFAs, которые рассматриваются в этой статье, являются сокращенными.

2. Алгебраический метод проверки эквивалентности автоматов

Алгебраический метод проводит проверку эквивалентности двух машин (автоматов, программ) в два этапа. На первом этапе формируется система уравнений, которая служит алгебраической спецификацией задачи проверки эквивалентности двух заданных машин. Выбор алгебры, в которой может быть построена подходящая система уравнений, зависит, прежде всего, от операционной семантики того типа машин, для которого изучается проблема эквивалентности. На втором этапе проверяется разрешимость построенной системы уравнений. Проверка проводится путем применения равносильных преобразований; эти преобразования либо приводят проверяемую систему к заранее выделенной приведенной форме, которая наверняка имеет решение, либо обнаруживают присутствие в системе несовместных уравнений, которые свидетельствуют о том, что такая система не имеет решений. Применяемые равносильные преобразования могут учитывать специфические особенности устройства проверяемых машин, такие как детерминированность, однозначность, префиксность и др.

Основные положения этого метода объясним на примере решения проблемы эквивалентности для детерминированных конечных автоматов: для заданного DFA A и пары его состояний p и q проверить эквивалентность этих состояний автомата. Два состояния p и q некоторого DFA $A = \langle Q, q_0, F, \varphi \rangle$ считаются эквивалентными, если для DFA $A(p) = \langle Q, p, F, \varphi \rangle$ и $A(q) = \langle Q, q, F, \varphi \rangle$, имеющих состояния p и q в качестве начальных, верно равенство $L(A(p)) = L(A(q))$.

Пусть задан DFA $A = \langle Q, q_0, F, \varphi \rangle$, работающий в алфавите Σ . Для проверки эквивалентности пары состояний p и q этого автомата построим систему уравнений $\mathcal{E}(A, p, q) = \mathcal{E}(A) \cup \{X_p = X_q\}$, состоящую из двух частей. Подсистема $\mathcal{E}(A)$ представляет собой спецификацию DFA A в алгебре регулярных выражений, а уравнение $X_p = X_q$ выражает требование эквивалентности состояний p и q . Эту систему уравнений можно составить и эффективно проверить ее разрешимость для любого DFA, однако, для упрощения изложения далее будем полагать, что рассматриваемый DFA A является сокращенным, т.е. из любого его состояния достижимо некоторое допускающее состояние.

Для построения системы уравнений $\mathcal{E}(A)$ каждому состоянию $r \in Q$ поставим в соответствие индивидуальную переменную X_r и константу δ_r , которая равна 1, если r — допускающее состояние, и 0 в противном случае. Уравнения системы $\mathcal{E}(A)$ формируются из регулярных выражений (р-выражений). Значения р-выражений — это языки в алфавите Σ . Р-выражения строятся из переменных X_r , соответствующих состояниям $r \in Q$, и констант при помощи операций конкатенации $\cdot$ и альтернации $+$; константами являются 0, 1, а также буквы алфавита Σ . Значениями констант 0 и 1 являются соответственно пустой язык $L_0 = \emptyset$ и язык $L_1 = \{\varepsilon\}$, состоящий только из пустого слова, а значением каждой константы $a \in \Sigma$ является язык $L_a = \{a\}$, состоящий только из однобуквенного слова a . Мы будем неявно использовать тождества алгебры р-выражений: $0 \cdot H = H \cdot 0 = 0$, $0 + H = H + 0 = H$.

Система уравнений (1) хорошо известна в теории автоматов — она используется в некоторых алгоритмах построения регулярных выражений для языков, допускаемых DFA (см., например, [67]). Она имеет вид

$$\mathcal{E}(A) = \{X_r = \sum_{a \in \Sigma} a \cdot H_{r,a} + \delta_r : r \in Q\}, \quad (1)$$

$$\text{где } H_{r,a} = \begin{cases} X_{\varphi(r,a)}, & \text{если значение функции переходов } \varphi(r, a) \text{ определено,} \\ 0, & \text{в противном случае.} \end{cases}$$

Как показано в учебнике [67], для любого сокращенного DFA A система уравнений (1) имеет единственное решение $\{X_r = L(A(r)) : r \in Q\}$, причем все языки $L(A(r))$ непусты. Отсюда следует, что состояния p и q эквивалентны тогда и только тогда, когда система уравнений $\mathcal{E}(A, p, q)$ имеет решение. Так задача проверки эквивалентности состояний DFA сводится к задаче проверки разрешимости системы линейных уравнений в алгебре р-выражений.

Решение системы линейных уравнений $\mathcal{E}(A, p, q)$ проводится традиционным для линейной алгебры методом Гаусса полного исключения переменных. В случае применения этого метода для проверки эквивалентности состояний DFA системы уравнений $\mathcal{E}_t$, которые образуются на каждой итерации t , состоят из двух частей: первая часть — это подсистема базовых уравнений $\mathcal{E}(A')$ для некоторого сокращенного DFA A' (этот автомат может быть отличен от исходного заданного DFA A), а вторая часть — это множество проверочных уравнений вида $X_i = X_j$, в обеих частях которых присутствуют только переменные. Проверочное уравнение $X_i = X_j$ из системы уравнений $\mathcal{E}_t$ указанного вида и переменная X_i из левой части этого уравнения называются *приведенными*, если X_i не имеет других вхождений в системе. Система уравнений $\mathcal{E}$ называется *приведенной*, если все проверочные уравнения этой системы являются приведенными. Нетрудно заметить, что любая приведенная система уравнений имеет решение.

В неприведенной системе уравнений $\mathcal{E}_t$ переменная X_i из левой части хотя бы одного проверочного уравнения $X_i = X_j$ имеет другие вхождения в уравнения системы $\mathcal{E}_t$. Такую переменную X_i будем называть *активной* переменной этой системы уравнений. В частности, исходная система уравнений $\mathcal{E}_1 = \mathcal{E}(A, p, q)$, соответствующая задаче проверки эквивалентности состояний p, q заданного DFA A , не является приведенной, и переменная X_p из левой части проверочного уравнения $X_p = X_q$ — это активная переменная.

Проверка разрешимости системы уравнений $\mathcal{E}(A, p, q)$ проводится итеративно путем применения равносильных преобразований. Эти преобразования деактивизируют одну за другой переменные проверяемой системы, до тех пор пока не будет получена либо приведенная система уравнений, которая, как было отмечено, всегда имеет решение, либо система, содержащая уравнения вида $X_i = 0$ или $0 = 1$, которые несовместны с остальными уравнениями системы и, тем самым, свидетельствуют о неразрешимости этой системы.

В начале первой итерации полагаем $\mathcal{E}_1 = \mathcal{E}(A, p, q)$. На каждой итерации t в неприведенной системе уравнений $\mathcal{E}_t$ выбирается активная переменная, проводится ее деактивизация (“исключение”), и затем восстанавливается (если это необходимо) канонический вид системы уравнений $\mathcal{E}_{t+1}$. Правила равносильных преобразований на t -й итерации таковы.

- 1) *Удаление тождеств.* Уравнения вида $X = X$ удаляются из системы $\mathcal{E}_t$.
- 2) *Проверка активности.* Если в систем $\mathcal{E}_t$ нет активных уравнений, то $\mathcal{E}_t$ — приведенная система уравнений. Тогда процедура проверки разрешимости завершает свое выполнение и объявляет о разрешимости исходной системы уравнений $\mathcal{E}(A, p, q)$.
- 3) *Деактивизация переменных.* В системе $\mathcal{E}_t$ выбирается проверочное уравнение $X_i = X_j$, содержащее активную переменную X_i , и во всех остальных уравнениях системы $\mathcal{E}_t$ все вхождения переменной X_i замещаются переменной X_j за счет применения подстановки $\{X_i/X_j\}$.

После этого шага уравнение $X_i = X_j$ становится приведенным, и число приведенных переменных в системе $\mathcal{E}_t$ увеличивается на 1. Однако применение подстановки $\{X_i/X_j\}$ к уравнениям системы $\mathcal{E}_t$ может давать побочный эффект: два базовых уравнения вида

$$\begin{aligned} X_j &= \sum_{a \in \Sigma} a \cdot H'_{j,a} + \delta'_j \\ X_j &= \sum_{a \in \Sigma} a \cdot H''_{j,a} + \delta''_j \end{aligned}$$

с одной и той же переменной X_j в левых частях появляются в системе $\mathcal{E}_t$. Это отклонение системы уравнений от канонической формы должно быть устранено.

- 4) *Устранение дублирующих базовых уравнений.* Удалим из системы $\mathcal{E}_t$ одно из двух указанных выше уравнений с переменной X_j в левой части и добавим в систему $|\Sigma|$ уравнений $H'_{j,a} = H''_{j,a}$ для каждой буквы $a \in \Sigma$, а также уравнение $\delta'_j = \delta''_j$. Нетрудно заметить, что после применения этого преобразования будет получена равносильная система уравнений, в которой могут содержаться “нестандартные” уравнения вида $X_r = 0$ или $c_1 = c_2$, где c_1, c_2 — константы.

- 5) *Устранение “нестандартных” уравнений.* Если среди добавленных уравнений есть уравнения вида $X_r = 0$ или $0 = 1$, то процедура проверки завершает свое выполнение и объявляет о неразрешимости исходной системы уравнений $\mathcal{E}(A, p, q)$. Если таких уравнений нет, т.е. среди добавленных уравнений есть только одно “нестандартное” уравнение (тождество) вида $0 = 0$ или $1 = 1$, то оно удаляется из системы.

Если в результате применения указанных правил процедура проверки еще не завершила свою работу, то будет получена система уравнений $\mathcal{E}_{t+1}$, которая равносильна системе $\mathcal{E}_t$, но имеет большее число приведенных переменных. После этого процедура проверки переходит к следующей итерации $t + 1$.

Утверждение 1. Для любого сокращенного DFA A и пары его состояний p, q предложенная процедура после применения к системе уравнений $\mathcal{E}_1 = \mathcal{E}(A, p, q)$ завершает работу и дает правильный ответ на вопрос о разрешимости системы $\mathcal{E}_1$.

Доказательство. После каждой итерации t число разрешенных переменных в системе уравнений $\mathcal{E}_{t+1}$ увеличивается, по меньшей мере, на единицу, и поэтому процедура проверки обязательно завершает работу. Преобразования систем уравнений $\mathcal{E}_t$ по правилам 1) и 3)–5) являются равносильными преобразованиями, и поэтому на каждой итерации t система уравнений $\mathcal{E}_t$ равносильна исходной системе $\mathcal{E}(A, p, q)$. Так как приведенная система уравнений всегда имеет решение, заключение о разрешимости системы уравнений $\mathcal{E}(A, p, q)$, которое процедура принимает на шаге 2), является правильным. Так как для сокращенного DFA система уравнений $\mathcal{E}(A)$ имеет единственное решение $X_r = L(A(r)) \neq \emptyset$ для любого состояния $r \in Q$, заключение о неразрешимости системы уравнений $\mathcal{E}(A, p, q)$, которое процедура принимает на шаге 5), является правильным. $\square$

Таким образом, верна

Теорема 1. Проблема эквивалентности для сокращенных DFA разрешима за время $O(n \log n)$, где n — размер DFA.

Доказательство. Согласно утверждению 1 описанная выше разрешающая процедура корректно проверяет эквивалентность пары состояний любого DFA A . Число итераций этой процедуры не превосходит числа состояний этого автомата. Чтобы сократить вычислительные издержки, данные, с которыми работает процедура проверки разрешимости системы уравнений $\mathcal{E}(A, p, q)$, можно представить при помощи ссылочных структур. Например, множество вхождений каждой переменной в r -выражения системы уравнений $\mathcal{E}(A, p, q)$ можно задать в виде слабо сбалансированного дерева: корень дерева представляет имя переменной X_r , а листовые вершины используются для ссылки на них всех вхождений переменной X_r в r -выражения проверяемой системы уравнений. Две основные операции, которые выполняет процедура проверки разрешимости систем уравнений, — сравнение имен переменных и подстановка одной переменной вместо всех вхождений другой, — могут быть выполнены за время, линейное относительно высоты деревьев, представляющих эти переменные. Применение подстановки X/Y можно осуществить двумя способами:

1. Корень дерева, представляющего одну из этих переменных, присоединяется к одной из наиболее близких к корню листовых вершин дерева, представляющего другую переменную, и образованное в результате дерево получает имя X ;
2. Корни обоих деревьев, представляющих переменные X и Y , присоединяются к новому корню, и образованное в результате дерево получает имя X .

Так как общее число вхождений всех переменных в уравнения системы $\mathcal{E}(A, p, q)$ не превосходит величины $|\Sigma|n$, такое моделирование операции подстановки позволяет поддерживать во всех деревьях, представляющих вхождения переменных, высоту, не превосходящую величины $O(\log n)$. Таким образом, каждая итерация процедуры проверки разрешимости системы уравнений $\mathcal{E}(A, p, q)$ может быть выполнена за время, логарифмически зависящее от размера DFA A . $\square$

Хотя в утверждении 1 говорится о применении процедуры проверки разрешимости только к системам уравнений, порожденным сокращенными DFA, это ограничение не является принципиальным. Чтобы проверять разрешимость систем уравнений $\mathcal{E}(A, p, q)$ для произвольных DFA, достаточно оснастить описанную процедуру двумя дополнительными правилами равносильных преобразований:

б) *Подстановка константы 0.* Если система $\mathcal{E}_t$ содержит уравнение $X_i = 0$, то во всех остальных уравнениях системы $\mathcal{E}_t$ все вхождения переменной X_i замещаются константой 0 за счет применения подстановки $\{X_i/0\}$.

7) *Распространение константы 0*. Если система $\mathcal{E}_t$ содержит уравнение $0 = a_{i_1} \cdot X_{j_1} + \dots + a_{i_k} \cdot X_{j_k}$, то это уравнение удаляется из системы, и вместо него в систему добавляются уравнения $X_{j_1} = 0, \dots, X_{j_k} = 0$.

Наиболее важным для успешного применения описанного метода проверки разрешимости систем уравнений является правило устранения дублирующих уравнений (правило 4), которое восстанавливает канонический вид системы после применения подстановки. Правила такого вида существенно зависят от специфических особенностей устройства автоматов рассматриваемого класса. В случае конечных автоматов такой особенностью является детерминированность автомата. Благодаря этому свойству уравнение

$$a_{i_1} \cdot X_{j_1} + \dots + a_{i_k} \cdot X_{j_k} = a_{i_1} \cdot Y_{j_1} + \dots + a_{i_k} \cdot Y_{j_k}$$

распадается на серию более простых уравнений $X_{j_1} = Y_{j_1}, \dots, X_{j_k} = Y_{j_k}$, в левых частях которых стоят переменные, и система уравнений вновь принимает канонический вид. Например, для уравнений, описывающих недетерминированный конечный автомат, такого расщепления сделать невозможно. Поэтому метод исключения переменных применим для проверки разрешимости систем линейных языковых уравнений, которые порождены лишь такими автоматами (машинами, программами), которые обладают некоторой “однозначностью” поведения. Одним из проявлений этой особенности поведения автоматов является свойство префиксности тех языков, которые фигурируют в системах уравнений, описывающих работу автомата. Изучению этого свойства посвящен следующий раздел статьи.

3. Префиксные языки и их основные свойства

Предложенный в этой статье метод проверки эквивалентности конечных автоматов различных типов существенно опирается на некоторые простые свойства семейств попарно вполне несовместных префиксных регулярных языков. Язык L называется *префиксным*, если все его слова попарно несовместны. Согласно приведенному определению пустой язык $\emptyset$ является префиксным, и язык $L = \{\varepsilon\}$ — это единственный префиксный язык, содержащий пустое слово ε . Два языка L_1 и L_2 называются *совместными*, если каждое слово из одного языка совместно с некоторым словом из другого. Будем говорить, что язык L_1 *не имеет расширений* в языке L_2 , если ни одно слово из языка L_2 не является расширением каких-либо слов из языка L_1 . Языки L_1 и L_2 назовем *вполне несовместными* ни один из них не имеет расширений в другом, т.е. любые два слова $w_1 \in L_1$ и $w_2 \in L_2$ несовместны.

Сформулированные ниже утверждения следуют непосредственно из определений свойств несовместности, префиксности и отсутствия расширений в языках. Они являются ключевыми при построении эффективных разрешающих алгоритмов в разделах 5, 6, и 8.

Утверждение 2. Если язык L_1 не имеет расширений в языке L_2 , то для любого слова w и языка L верно, что

- язык L_1 не имеет расширений в языке L_2/w ,
- язык $L_1 \setminus w$ не имеет расширений в языке $L_2 \setminus w$ и
- язык $L_1 L$ не имеет расширений в языке L_2 .

Утверждение 3. Если языки L_1 и L_2 не имеют расширений в обоих языках L_3 и L_4 , то $L_1 \cup L_2$ не имеет расширений в языке $L_3 \cup L_4$.

Утверждение 4. Если языки L_1 и L_2 вполне несовместны, то для любого языка L языки $L_1 L$ и L_2 также вполне несовместны.

Утверждение 5. Для любых префиксных языков L_1 и L_2 , и всякого слова w языки $L_1 L_2$ и $L_1 \setminus w$ являются префиксными.

Утверждение 6. Для любой пары вполне несовместных префиксных языков L_1 и L_2 язык $L_1 \cup L_2$ является префиксным.

Утверждение 7. Если язык L_1 является префиксным, а язык L_2 не имеет расширений в языке L_3 , то язык $L_1 L_2$ также не имеет расширений в языке $L_1 L_3$.

Утверждение 8. Для любой пары вполне несовместных префиксных языков L_1 и L_2 язык $L_1^* L_2$ является префиксным.

Доказательство. Возьмем пару вполне несовместных префиксных языков L_1 и L_2 .

1. Если $\varepsilon \in L_1$, то $L_2 = \emptyset$, поскольку пустое слово ε совместно с любым словом. Значит, язык $L_1^* L_2 = \emptyset$ является префиксным.

2. Рассмотрим случай $\varepsilon \notin L_1$. Предположим, что язык $L_1^* L_2$ не является префиксным, и пусть слово $w' = u'_1 \dots u'_k v'$, где $\{u'_1, \dots, u'_k\} \subseteq L_1$, $v' \in L_2$, — это кратчайшее слово из множества $L_1^* L_2$, которое совместно с некоторым другим словом $w'' = u''_1 \dots u''_m v''$ из множества $L_1^* L_2$, где $\{u''_1, \dots, u''_m\} \subseteq L_1$, $v'' \in L_2$.

Если $k > 0$, то непустое слово u'_1 может быть совместно либо со словом v'' (в случае $m = 0$), либо со словом u''_1 (в случае $m > 0$). Первая альтернатива невозможна, поскольку языки L_1 и L_2 вполне несовместны. Так как язык L_1 является префиксным, вторая альтернатива возможна только тогда, когда $u'_1 = u''_1$. Но отсюда следует, что более короткие слова $u'_2 \dots u'_k v'$ и $u'_2 \dots u'_m v''$ также совместны вопреки выбору слова w' .

Если $k = 0$, то v' совместно либо со словом u''_1 (в случае $m > 0$), либо со словом v'' (в случае $m = 0$). Первая альтернатива невозможна, поскольку языки L_1 и L_2 вполне несовместны. А поскольку язык L_2 является префиксным, вторая альтернатива возможна лишь при условии $v' = v''$. Отсюда следует $w' = w''$, что противоречит выбору слова w' .

Полученные противоречия приводят к заключению о том, что язык $L_1^* L_2$ является префиксным. $\square$

Утверждение 9. Если префиксный язык L_1 вполне несовместен с языком L_2 , и при этом оба языка L_1 и L_2 не имеют расширений в языке L_3 , то язык $L_1^* L_2$ также не имеет расширений в языке $L_1^* L_3$.

Доказательство утверждения 9 проводится по той же схеме *reductio ad absurdum*, что и доказательство утверждения 8.

Таким образом, некоторые теоретико-языковые операции, такие как объединение, конкатенация, деление слева и справа, сохраняют свойства несовместности, префиксности и отсутствия расширений в языках.

Утверждение 10. Пусть L' и L'' — конечные непустые префиксные совместные языки. Тогда существует единственное разбиение $L' = \bigcup_{i=1}^n L'_i$ и $L'' = \bigcup_{i=1}^n L''_i$ этих двух языков, при котором для каждого $i, 1 \leq i \leq n$, одно из подмножеств L'_i или L''_i состоит из одного слова $\{w\}$, а все слова из другого подмножества являются расширениями этого слова w .

Доказательство. Воспользуемся индукцией по числу слов в множестве $L' \cup L''$. Базис индукции — случай, когда $|L'| = 1$ или $|L''| = 1$, — очевиден. Для обоснования индуктивного перехода рассмотрим кратчайшее слово w в множестве $L' \cup L''$. Предположим для определенности, что $w \in L'$. Тогда возьмем множество слов $\{w\}$ в качестве L'_1 и все слова из L'' , которые совместны со словом w , в качестве L''_1 . Так как w — это одно из кратчайших слов множества $L' \cup L''$, а L' и L'' — совместные языки, множество слов L''_1 непусто и состоит только из расширений слова w . Так как язык L' является префиксным, ни одно слово из L'_1 не является расширением или префиксом какого-либо

слова из множества $L' - \{w\}$. Таким образом, пара подмножеств L'_1 и L''_1 удовлетворяют требованиям утверждения 10 и определяется однозначно выбором слова $w \in L'_1$. Множества $L' - \{w\}$ и $L'' - L'_1$ — это непустые префиксные и совместные языки; по индуктивному предположению они имеют единственное разбиение, удовлетворяющее требованиям утверждения 10. $\square$

Такое разбиение пары совместных префиксных языков $L' = \bigcup_{i=1}^n L'_i$ и $L'' = \bigcup_{i=1}^n L''_i$ будем называть *расщеплением* языков L' и L'' . Пары соответствующих друг другу подмножеств L'_i и L''_i , $1 \leq i \leq n$, такого расщепления будут называться *долями*. Утверждение 10 сыграет ключевую роль в построении алгоритма проверки эквивалентности трансдюсеров реального времени в разделе 5.

Обозначим записью $PFR\text{eg}(\Sigma)$ семейство всех префиксных регулярных языков. Конечные автоматы, которые описывают префиксные языки, обладают важным характерным свойством, которое существенно облегчает работу с этими автоматами.

Утверждение 11 ([68, 69]). Пусть $A = \langle Q, q_0, F, \varphi \rangle$ — сокращенный DFA в алфавите Σ . Тогда регулярный язык $L(A)$ является префиксным тогда и только тогда, когда значение функции переходов $\varphi(q, x)$ не определено для любого допускающего состояния q из F и любой буквы x из Σ , т.е. из допускающих состояний автомата A не исходит никаких переходов.

Как следует из утверждения 11, если язык $L(A)$ является префиксным, то все допускающие состояния автомата A можно безопасно объединить в одно единственное допускающее тупиковое состояние f , у которого нет исходящих из него переходов. Автомат $A = \langle Q, q_0, \{f\}, \varphi \rangle$ такого вида будет называться *префиксным DFA*. Аналогично, каждое семейство регулярных языков $\mathcal{F} = \{L_1, \dots, L_m, L_{m+1}\}$, в котором $L_1, \dots, L_m$ — это попарно вполне несовместные префиксные языки, которые не имеют расширений в языке L_{m+1} , можно описать при помощи multi-DFA $D = \langle Q, q_0, \{f_1\}, \dots, \{f_m\}, F_{m+1}, \varphi \rangle$, в котором допускающие состояния $f_1, \dots, f_m$ не имеют исходящих из них переходов. Такой multi-DFA, описывающий семейство языков $\mathcal{F}$, будет называться *префиксным multi-DFA*.

В разрешающих алгоритмах, представленных в разделах 5, 6, и 8, будут применяться пять операций над регулярными языками: деление регулярного языка на слово слева $L \setminus w$ и справа L/a , объединение языков $L_1 \cup L_2$, конкатенация языков $L_1 L_2$, и комбинированная итерация языков $L_1^* L_2$. Префиксные DFA и multi-DFA, используемые для описания префиксных регулярных языков, предоставляют значительные преимущества для эффективного выполнения этих операций.

Для заданного слова $w = x_1 \dots x_n$ и multi-DFA $D = \langle Q, q_0, F_1, \dots, F_m, \varphi \rangle$, который описывает семейство регулярных языков $\{L_1, \dots, L_m\}$, достаточно всего лишь отыскать w -последователя q' начального состояния q_0 (т.е. такое состояние q' , для которого имеется последовательность переходов $q_0 \xrightarrow{x_1} \dots \xrightarrow{x_n} q'$), чтобы получить multi-DFA representation $D' = \langle Q, q', F_1, \dots, F_m, \varphi \rangle$, описывающий семейство языков $\{L_1 \setminus w, \dots, L_m \setminus w\}$.

Для заданной буквы $a \in \Sigma$ и DFA $A = \langle Q, q_0, F, \varphi \rangle$, нетрудно отыскать подмножество всех a -предшественников $F' = \{q' : q' \xrightarrow{a} q, q \in F\}$ допускающих состояний и построить DFA $A' = \langle Q, q_0, F', \varphi \rangle$, описывающий частное от деления $L(A)/a$.

Если multi-DFA $D = \langle Q, q_0, F_1, F_2, \varphi \rangle$ описывает пару регулярных языков L_1 и L_2 , то их объединение $L_1 \cup L_2$ можно описать при помощи DFA $B' = \langle Q, q_0, F_1 \cup F_2, \varphi \rangle$.

Объединив допускающее состояние префиксного DFA, описывающего язык L_0 , и начальное состояние multi-DFA, который описывает семейство языков $L_1, \dots, L_m$, можно построить автоматное описание семейства конкатенаций $\{L_0 L_1, \dots, L_0 L_m\}$.

И, наконец, если имеется префиксный multi-DFA $D = \langle Q, q_0, \{f_1\}, \dots, \{f_m\}, F, \varphi \rangle$, то 1) добавив к нему копию q'_0 начального состояния q_0 вместе с исходящими из него переходами, 2) перенаправив в состояние q'_0 все те переходы, которые ранее вели в состояние q_0 , и 3) объединив начальное состояние q_0 и допускающее состояние f_1 , можно построить multi-DFA $D' = \langle Q \cup \{q'_0\}, q_0, \{f_2\}, \dots, \{f_m\}, F, \varphi' \rangle$, который описывает семейство языков $L_1^* L_2, \dots, L_1^* L_m, L_1^* L_{m+1}$.

Все эти операции над языками можно выполнить за время, линейно зависящее от размера заданных автоматных описаний языков, и, кроме того, размер (число состояний) полученных при этом автоматов увеличивается не более чем на 1.

4. Конечные автоматы-преобразователи

Конечный автомат-преобразователь (далее коротко, трансдюсер), работающий в конечном входном алфавите Σ и конечном выходном алфавите Δ , представляет собой систему переходов $\pi = \langle Q, F, \longrightarrow \rangle$, в которой Q — конечное множество *состояний*, $F \subseteq Q$ — подмножество *финальных состояний*, и $\longrightarrow$ — конечное *отношение переходов* типа $Q \times (\Sigma \cup \{\varepsilon\}) \times \Delta^* \times Q$. Иногда для обозначения трансдюсеров будет использоваться запись $\pi(q_0)$, чтобы выделить некоторое состояние q_0 из множества Q как *начальное состояние* π . Четверки (q, x, u, q') из отношения $\longrightarrow$ называются *переходами* и обозначаются записью $q \xrightarrow{x|u} q'$. Если x — это буква алфавита Σ , то переход $q \xrightarrow{x|u} q'$ будем называть Σ -переходом. Переходы вида $q \xrightarrow{\varepsilon|u} q'$ называются ε -переходами. Нетрудно понять, что Σ -переход $q \xrightarrow{a|u} q'$ означает, что трансдюсер, пребывающий в состоянии q , считывает букву a на входной ленте, записывает на выходной ленте слово u и переходит в состояние q' . Когда в трансдюсере срабатывает ε -переход $q \xrightarrow{\varepsilon|u} q'$, то машина, не обращаясь к входной ленте, записывает слово u на выходной ленте и переходит в состояние q' . Размером перехода $q \xrightarrow{x|u} q'$ будет считаться число $|u| + 1$. Размером $|\pi|$ трансдюсера π является число, равное сумме размеров всех переходов машины π плюс число ее состояний $|Q|$.

Прогоном трансдюсера π из состояния q в состояние q' на входном слове h называется всякая конечная последовательность переходов

$$q \xrightarrow{x_1|u_1} q_1 \xrightarrow{x_2|u_2} \dots \xrightarrow{x_{n-1}|u_{n-1}} q_{n-1} \xrightarrow{x_n|u_n} q', \quad (2)$$

для которой выполняется равенство $h = x_1 x_2 \dots x_n$. Пара слов (h, u) , в которой $u = u_1 u_2 \dots u_n$, называется *пометкой* прогона (2). Мы будем использовать запись $q \xrightarrow{h|u} q'$ для обозначения того факта, что трансдюсер π имеет прогон (2) с пометкой (h, u) из состояния q в состояние q' . Если $q' \in F$, то прогон (2) называется *финальным*. Прогон (2) трансдюсера π с пометкой (ε, u) называется ε -прогоном.

Для каждого состояния q существует *пустой прогон* $q \xrightarrow{\varepsilon|\varepsilon} q$ из состояния q в то же самое состояние, представляющий собой пустую последовательность переходов.

Множество пар слов $TR(\pi, q) = \{(h, u) : q \xrightarrow{h|u} q', q' \in F\}$, которыми помечены все финальные прогоны машины π из состояния q , называется *отношением трансдукции*, *вычисляемое трансдюсером π в состоянии q* . Заметим, что если q — это финальное состояние трансдюсера π , отношение трансдукции $TR(\pi, q)$ содержит пометку $(\varepsilon, \varepsilon)$. Состояния q_1 и q_2 трансдюсера π называются *эквивалентными* (и этот факт обозначается записью $\pi(q_1) \sim \pi(q_2)$), если верно равенство $TR(\pi, q_1) = TR(\pi, q_2)$. Проблема эквивалентности для трансдюсеров состоит в том, чтобы для произвольного заданного трансдюсера π и пары его состояний q_1 и q_2 проверить выполнимость отношения $\pi(q_1) \sim \pi(q_2)$.

Автомат-преобразователь π называется *трансдюсером реального времени*, если он не содержит ε -переходов, т.е. $\rightarrow \subseteq Q \times \Sigma \times \Delta^* \times Q$. В отличие от конечных автоматов, в некоторых трансдюсерах невозможно избавиться от ε -переходов, и поэтому, вычислительные возможности трансдюсеров реального времени существенно уступают возможностям трансдюсеров с ε -переходами.

Трансдюсер реального времени π называется

- *детерминированным* трансдюсером, если для каждой входной буквы a и состояния q он имеет не более одного перехода вида $q \xrightarrow{a|u} q'$;
- *функциональным* (или *однозначным*) трансдюсером, если для каждого состояния q отношение трансдукции $TR(\pi, q)$ является функцией;
- *k -неоднозначным* трансдюсером, если для каждого состояния q и любого входного слова h имеется не более k финальных прогонов машины π из состояния q на слове h ;
- *k -значным* трансдюсером, если для каждого состояния q и любого входного слова h отношение $TR(\pi, q)$ содержит не более k образов слова h , т.е. $|Im(TR(\pi, q), h)| \leq k$;
- трансдюсером *ограниченного диапазона длин k* , если для каждого состояния q и любого входного слова h , множество длин образов w слова h в отношении $TR(\pi, q)$ содержит не более k чисел, т.е. $|\{|w| : w \in Im(TR(\pi, q), h)\}| \leq k$.

Для всех перечисленных выше типов трансдюсеров реального времени была доказана разрешимость проблемы эквивалентности (см. [21–24, 28, 35]). В следующем разделе вводится в рассмотрение новый класс трансдюсеров реального времени, для которого эта проблема также оказывается разрешимой.

5. Проверка эквивалентности префиксных трансдюсеров реального времени

Рассмотрим произвольный трансдюсер реального времени $\pi = \langle Q, q_0, F, \rightarrow \rangle$. Для любой пары состояний q, q' и входной буквы x обозначим записью $Out_\pi(q, q', x)$ множество всех выходных слов u на переходах вида $q \xrightarrow{x|u} q'$ в системе π . Трансдюсер π назовем *префиксным*, если для любого его состояния q , любой входной буквы x , и произвольной пары различных переходов вида $q \xrightarrow{x|u'} q'$ и $q \xrightarrow{x|u''} q''$ выходные слова u' и u'' несовместны. Префиксные трансдюсеры обладают очевидным характеристическим свойством: для каждого состояния $q \in Q$ и любой входной буквы $x \in \Sigma$ все языки семейства $\{Out_\pi(q, x, p) : p \in Q\}$ являются префиксными и попарно вполне несовместными.

Очевидно, что каждый детерминированный трансдюсер является префиксным, хотя обратное утверждение неверно. Префиксные трансдюсеры обладают особым свойством “однозначности”: для каждого состояния q префиксного трансдюсера π и любой пары $(h, u) \in TR(\pi, q)$ существует единственный прогон машины π из состояния q с пометкой (h, u) . Однако следует иметь в виду, что это свойство существенно отличается от свойства однозначности автоматов-преобразователей в его традиционном смысле (см. [24, 25, 27, 28, 31]): в обычном понимании свойство однозначности (ambiguity) требует, чтобы для каждого входного слова h существовало не более одного финального прогона трансдюсера π помеченного парой (h, w) для некоторого выходного слова w . Необходимо также заметить, что некоторые префиксные трансдюсеры реального времени не обладают такими свойствами как k -неоднозначность, k -значность или даже ограниченность диапазона длин, т.е. класс префиксных трансдюсеров не подпадает ни под один из ранее известных случаев разрешимости проблемы эквивалентности для трансдюсеров реального времени.

Пример 1. Рассмотрим простой префиксный трансдьюсер реального времени $\pi = \langle \{q_0, q_1\}, \{q_1\}, \rightarrow \rangle$, который работает в алфавитах $\Sigma = \{a, b\}$ и $\Delta = \{c, d\}$ и имеет три перехода $q_0 \xrightarrow{a/c} q_0$, $q_0 \xrightarrow{a/dd} q_0$, и $q_0 \xrightarrow{b/\varepsilon} q_1$. Для каждого входного слова $h_n = a^n b$ длины выходных слов на финальных прогонах трансдьюсера π из состояния q_0 являются всевозможными целыми числами в диапазоне от n до $2n$. Следовательно, трансдьюсер π не обладает свойством ограниченного диапазона длин.

Предложенный в этой статье метод проверки эквивалентности для префиксных трансдьюсеров реального времени основан на преобразованиях регулярных выражений специального вида. Точно также, как это было сделано для проверки эквивалентности DFAs в разделе 1, с каждым трансдьюсером π будем ассоциировать систему линейных уравнений $\mathcal{E}_\pi$, которая специфицирует трансдукции, вычисляемые во всех состояниях машины π . Чтобы проверить эквивалентность $\pi(q') \sim \pi(q'')$ добавим к системе уравнений $\mathcal{E}_\pi$ требование эквивалентности состояний, которое представлено уравнением вида $X' = X''$, а затем попробуем выяснить, имеет ли решение полученная система уравнений. Для этого, как и в случае проверки эквивалентности конечных автоматов, воспользуемся методом исключения Гаусса. Для произвольных трансдьюсеров преимущества такого подхода сомнительны, но для префиксных трансдьюсеров утверждение 10 о расщеплении совместных пар префиксных языков предоставляет удобное средство для эффективного решения систем уравнений, соответствующих задаче проверки эквивалентности.

Чтобы упростить изложение этого метода, сделаем некоторые предположения относительно преобразователя, подлежащего анализу:

- входной алфавит $\Sigma = \{a_1, \dots, a_k\}$ и выходной алфавит Δ не имеют общих букв; символы x, y, z будут использоваться для обозначения произвольных букв из множества Σ , а символы u, v, w — для обозначения произвольных слов из множества Δ^* ,
- мы будем рассматривать только *сокращенные* трансдьюсеры π , у которых из каждого состояния достижимо хотя бы одно финальное состояние.

Регулярные выражения (далее, для краткости, r -выражения) строятся из переменных $X_1, X_2, \dots$, констант $0, 1$, букв из множества Σ , и слов из множества Δ^* при помощи операций конкатенации $\cdot$ и альтернатиции $+$. R -выражения интерпретируются в полукольце трансдукций. Значениями констант $0, 1$, каждой буквы $x \in \Sigma$, и каждого слова $u \in \Delta^*$ служат соответственно трансдукции $\emptyset, \{(\varepsilon, \varepsilon)\}, \{(x, \varepsilon)\}$ и $\{(\varepsilon, u)\}$. Конкатенация $\cdot$ и альтернатиция $+$ истолковываются соответственно как конкатенация и объединение трансдукций. Очевидно, что операция альтернатиции $+$ коммутативна, и равенство $x \cdot u = u \cdot x$ верно для любой буквы $x \in \Sigma$ и слова $u \in \Delta^*$. Мы также будем использовать следующие очевидные тождества: $0 \cdot T = T \cdot 0 = 0, 0 + T = T, 1 \cdot T = T \cdot 1 = T$.

В центре внимания будут линейные r -выражения двух типов. Δ -выражением будем называть константу 0 , а также всякое r -выражение вида $u_1 \cdot X_{i_1} + u_2 \cdot X_{i_2} + \dots + u_m \cdot X_{i_m}$, в котором одна и та же переменная может иметь несколько вхождений. Если слова $u_1, u_2, \dots, u_m$ попарно несовместны, то такое Δ -выражение называется префиксным. Σ -выражением называется всякое выражение вида $a_1 \cdot E_1 + a_2 \cdot E_2 + \dots + a_k \cdot E_k + w_1 + \dots + w_m$, в котором $E_i, 1 \leq i \leq k$, — это Δ -выражения, и при этом входной алфавит $\Sigma = \{a_1, a_2, \dots, a_k\}$ и $\{w_1, \dots, w_m\} \subseteq \Sigma^*$. Мы будем использовать записи $E(X_1, \dots, X_n)$ (для Δ -выражений) или $G(X_1, \dots, X_n)$ (для Σ -выражений), чтобы подчеркнуть тот факт, что эти r -выражения могут содержать только переменные $X_1, \dots, X_n$.

Рассмотрим произвольный трансдьюсер реального времени $\pi = \langle Q, F, \rightarrow \rangle$, работающий в алфавитах Σ и Δ . С каждым его состоянием p будем ассоциировать переменную X_p , и для каждой пары $p \in Q$ и $x \in \Sigma$ построим Δ -выражение $E_{p,x} = \sum_{q \in Q} \sum_{u \in \text{Out}_\pi(p,x,q)} u \cdot X_q$. Ясно, что если трансдьюсер π является префиксным, то все определенные таким образом Δ -выражения $E_{p,x}$ также являются префиксными. Для каждого состояния p обозначим записью δ_p либо константу 1 в случае $p \in F$,

или константу 0 в противном случае. Тогда алгебраической спецификацией трансдьюсера π может служить система уравнений

$$\mathcal{E}_\pi = \{X_p = \sum_{x \in \Sigma} x \cdot E_{p,x} + \delta_p : p \in Q\}.$$

Утверждение 12. Для любого трансдьюсера реального времени π система уравнений $\mathcal{E}_\pi$ имеет единственное решение $\{X_p = TR(\pi, p) : p \in Q\}$.

Доказательство. То, что множество трансдукций $TR(\pi, p)$, $p \in Q$, взятых в качестве значений соответствующих переменных X_p , — это одно из решений системы уравнений $\mathcal{E}_\pi$, следует непосредственно из определения отношений $TR(\pi, p)$ и устройства уравнений этой системы.

Чтобы доказать единственность этого решения, мы рассмотрим произвольное решение $X_p = T_p$, $p \in Q$, системы уравнений $\mathcal{E}_\pi$ и покажем при помощи индукции по длине входного слова h , что для любой пары (h, w) и состояния $p \in Q$ пара (h, w) принадлежит отношению T_p тогда и только тогда, когда $(h, w) \in TR(\pi, p)$.

Как можно заметить из описания уравнений системы $\mathcal{E}_\pi$, пара (ε, w) может содержаться в T_p в том и только том случае, если $w = \varepsilon$ и $p \in F$; последнее равносильно включению $(\varepsilon, w) \in TR(\pi, p)$.

Предположим, что соотношение $(h, w) \in T_p \iff (h, w) \in TR(\pi, p)$ выполняется для каждого состояния p , любого выходного слова w , и произвольного входного слова h , длина которого не превосходит n . Рассмотрим произвольное входное слово $h' = yh$ длины $n + 1$. Так как отношение T_p является составной частью решения системы уравнений $\mathcal{E}_\pi$, верно равенство $T_p = \sum_{x \in \Sigma} \sum_{q \in Q} \sum_{u \in Out_\pi(p, x, q)} x \cdot u \cdot T_q + c_p$.

Поэтому для любого выходного слова w пара (h', w) содержится в отношении T_p тогда и только тогда, когда $(h, w \setminus u) \in T_q$ для некоторого состояния q и выходного слова $u \in Out_\pi(p, y, q)$. Согласно индуктивному предположению последнее условие равносильно тому, что $(h, w \setminus u) \in TR(\pi, q)$, и при этом в трансдьюсере π имеется переход $p \xrightarrow{y|u} q$; это означает, что $(h, w) \in TR(\pi, p)$. $\square$

Следствие 1. Для каждой пары состояний $p, q \in Q$ эквивалентность $\pi(p) \sim \pi(q)$ имеет место тогда и только тогда, когда система уравнений $\mathcal{E}_\pi \cup \{X_p = X_q\}$ имеет решение.

Как показывает полученное следствие, для проверки эквивалентности трансдьюсеров реального времени достаточно уметь эффективно проверять разрешимость систем уравнений, которые являются расширением систем уравнений $\mathcal{E}_\pi$, соответствующих трансдьюсерам. Разрешимость некоторых расширений совершенно очевидна. Будем говорить, что система линейных уравнений

$$\mathcal{E} = \mathcal{E}_\pi(X_1, \dots, X_n) \cup \{Y_j = E_j(X_1, \dots, X_n) : 1 \leq j \leq m\},$$

является *приведенной*, если множества переменных $\{X_1, \dots, X_n\}$ и $\{Y_1, \dots, Y_m\}$ не пересекаются, и правые части E_j , $1 \leq j \leq m$, — это некоторые r -выражения.

Утверждение 13. Каждая приведенная система уравнений $\mathcal{E}$ имеет единственное решение.

Доказательство. Следует непосредственно из утверждения 12. $\square$

Некоторые другие расширения систем уравнений $\mathcal{E}_\pi$ не имеют решений.

Утверждение 14. Если система уравнений $\mathcal{E}_\pi(X_1, \dots, X_n) \cup \{\sum_{i=1}^{\ell} u_i \cdot X_i = \sum_{j=1}^m v_j \cdot X_j\}$ имеет решение, то языки $L_1 = \{u_1, \dots, u_\ell\}$ и $L_2 = \{v_1, \dots, v_m\}$ совместны.

Доказательство. Согласно утверждению 12 система уравнений $\mathcal{E}_\pi$ имеет ненулевое решение $X_1 = T_1, \dots, X_n = T_n$. Если слово u_i несовместно со словами $v_1, \dots, v_m$, то $(\varepsilon, u_i)T_i \cap \bigcup_{j=1}^m (\varepsilon, v_j)T_j = \emptyset$, и поэтому $\sum_{i=1}^{\ell} u_i \cdot T_i \neq \sum_{j=1}^m v_j \cdot T_j$ □

Утверждение 15. Если система уравнений $\mathcal{E}_\pi(X_1, \dots, X_n) \cup \{X_1 = \sum_{i=1}^{\ell} u_i \cdot X_i\}$ имеет решение, и множество слов $\{u_1, \dots, u_\ell\}$ является префиксным, то $\ell = 1$ и $u_1 = \varepsilon$, т.е. $\sum_{i=1}^{\ell} u_i \cdot X_i \equiv X_1$.

Доказательство. Согласно утверждению 12 система уравнений $\mathcal{E}_\pi$ имеет единственное решение $X_1 = T_1, \dots, X_n = T_n$ и при этом $T_i \neq \emptyset$ для каждого $i, 1 \leq i \leq n$. Предположим, что верно равенство $T_1 = \bigcup_{i=1}^{\ell} (\varepsilon, u_i)T_i$. Рассмотрим произвольное входное слово h из области определения $Dom(T_1)$ и конечные языки из множеств значений $Im(T_i, h), 1 \leq i \leq \ell$. С одной стороны, мы имеем очевидное соотношение

$$T_1 = \bigcup_{i=1}^{\ell} (\varepsilon, u_i)T_i \implies Im(T_1, h) = \bigcup_{i=1}^{\ell} u_i Im(T_i, h).$$

С другой стороны, поскольку множество слов $\{u_1, \dots, u_\ell\}$ является префиксным, языки $u_1 Im(T_1, h)$ и $\bigcup_{i=2}^{\ell} u_i Im(T_i, h)$ не имеют общих слов. Поэтому мощность множества $\bigcup_{i=2}^{\ell} u_i Im(T_i, h)$ равна 0. Однако из равенства $Im(T_1, h) = u_1 Im(T_1, h)$ следует $u_1 = \varepsilon$. А так как пустое слово ε совместно с любым словом, мы приходим к выводу о том, что $\ell = 1$. □

Здесь нужно особо отметить, что в приведенном доказательстве решающую роль сыграл тот факт, что π — это трансдюсер реального времени, и поэтому для каждого входного слова h трансдукции T_i , являющиеся решениями системы уравнений $\mathcal{E}_\pi$, имеют конечное число образов слова h . Когда в разделе 6 речь пойдет о трансдюсерах с ε -переходами, утверждением 15 уже нельзя будет воспользоваться.

Далее опишем итеративную процедуру проверки разрешимости системы уравнений $\mathcal{E}_1 = \mathcal{E}_\pi \cup \{X_p = X_q\}$ для произвольного префиксного трансдюсера π путем преобразования этой системы в равносильную приведенную форму. В начале каждой итерации t эта процедура имеет на входе систему уравнений вида

$$\mathcal{E}_t = \mathcal{E}_{\pi_t} \cup \{X_{j_i} = E_i : 1 \leq i \leq N_t\},$$

где

- π_t — это некоторый префиксный трансдюсер (не обязательно совпадающий с анализируемым трансдюсером π),
- $X_{j_i}, 1 \leq i \leq N_t$, — переменные (не обязательно попарно различные),
- $E_i, 1 \leq i \leq N_t$, — префиксные ненулевые Δ -выражения, зависящие только от тех переменных, которые содержатся в уравнениях подсистемы $\mathcal{E}_{\pi_t}$.

Уравнения подсистемы $\mathcal{E}_{\pi_i}$ будем называть *базовыми уравнениями*, а все остальные уравнения — *проверочными уравнениями*. Если переменная X_i из левой части проверочного уравнения $X_i = E_i$ не встречается где-либо еще в системе $\mathcal{E}_t$, то X_i будет называться *приведенной переменной*, а соответствующее уравнение — *приведенным уравнением*. Все остальные проверочные уравнения и переменные из их левых частей будут называться *активными*. Очевидно, что система уравнений $\mathcal{E}_t$, не содержащая активных переменных, является приведенной. Наш подход основан на идее метода исключения Гаусса и деактивизирует одну за другой все переменные системы $\mathcal{E}_t$ за счет применения подходящих равносильных преобразований к системе $\mathcal{E}_t$. Этот процесс завершается в двух случаях:

- в системе уравнений $\mathcal{E}_t$ отсутствуют активные переменные; тогда согласно утверждению 13, система $\mathcal{E}_t$ имеет решение, и отсюда следует $\pi(p) \sim \pi(q)$;
- одно из активных уравнений системы $\mathcal{E}_t$ противоречит заключению утверждения 14 или утверждения 15; тогда система $\mathcal{E}_t$ не имеет решений, и поэтому состояния p и q трансдюсера π могут быть сочтены неэквивалентными.

На t -й итерации следующие равносильные преобразования применяются к системе уравнений $\mathcal{E}_t$.

1) *Удаление тождеств*. Уравнения вида $X = X$ удаляются из системы $\mathcal{E}_t$.

2) *Проверка активности переменных*. Если в систем $\mathcal{E}_t$ нет активных уравнений, то процедура завершает свое выполнение и объявляет о разрешимости исходной системы уравнений в соответствии с утверждением 13.

3) *Приведение переменных*. Выбирается активное уравнение $X_i = E_i$ в $\mathcal{E}_t$. Если X_i входит в состав Δ -выражения E_i с ненулевым коэффициентом, то процедура завершает свое выполнение и объявляет о разрешимости исходной системы уравнений в соответствии с утверждением 15. В противном случае во всех остальных уравнениях системы $\mathcal{E}_t$ все вхождения переменной X_i замещаются r -выражением E_i за счет применения подстановки $\{X_i/E_i\}$.

После этого шага переменная X_i и уравнение $X_i = E_i$ становятся приведенными, а число активных переменных в системе $\mathcal{E}_t$ сокращается. Однако применение $\{X_i/E_i\}$ к уравнениям системы дает и побочный эффект: нестандартные уравнения одного из следующих видов

А) $E = G$, где E — это Δ -выражение, отличное от переменных, и G — это Σ -выражение,

В) $E' = E''$, где E', E'' — это Δ -выражения, отличные от переменных,

могут появиться в системе $\mathcal{E}_t$. Может случиться также, что

С) сразу несколько базовых уравнений вида $X = G$ с одной и той же переменной X в левых частях появляются в системе $\mathcal{E}_t$. Эти отклонения системы уравнений от установленной канонической формы должны быть устранены.

4) *Устранение нестандартных уравнений вида $E = G$* . Для этого

- в каждом уравнении вида $E(X_1, \dots, X_\ell) = G$ заменить все вхождения переменных $X_i, 1 \leq i \leq \ell$, в ненулевом Δ -выражении E на Σ -выражения G_i из правых частей базовых уравнений $X_i = G_i$ подсистемы $\mathcal{E}_{\pi_i}$, а затем привести полученное выражение $E(G_1, \dots, G_\ell)$ к стандартной форме Σ -выражения при помощи закона коммутативности $u \cdot x = x \cdot u$ для букв алфавитов Σ и Δ и закона дистрибутивности $T \cdot R + T \cdot S = T \cdot (R + S)$;
- для каждой пары базовых уравнений $X = G'$ и $X = G''$ с одинаковыми левыми частями заменить одно из этих уравнений на уравнение $G' = G''$.

После этого шага все уравнения вида $E = G$ исчезнут, и все базовые уравнения $X = G$ будут иметь попарно различные переменные в левых частях. Однако это достигается за счет того, что в системе появляются нестандартные уравнения вида

Д) $G' = G''$, где G', G'' — это Σ -выражения.

5) Устранение нестандартных уравнений вида $G' = G''$. Рассмотрим уравнение

$$\sum_{i=1}^k a_i \cdot E'_i + w'_1 + \dots + w'_m = \sum_{i=1}^k a_i \cdot E''_i + w''_1 + \dots + w''_n.$$

Если $\{w'_1, \dots, w'_m\} \neq \{w''_1, \dots, w''_n\}$, то процедура завершается и объявляет о неразрешимости системы. То же самое решение будет принято, если для некоторого i , $1 \leq i \leq k$, одно из Δ -выражений E'_i или E''_i равно 0, тогда как другое отлично от нуля. Легко видеть, что в обоих случаях левая и правая части этого уравнения оцениваются по-разному для любых непустых трансдукций, взятых в качестве значений переменных.

В противном случае удалить это уравнение из системы и подставить вместо него уравнение $E'_i = E''_i$, $1 \leq i \leq k$, где E'_i и E''_i — ненулевые Δ -выражения. Таким образом, все уравнения вида $G' = G''$ исчезнут из системы, а вместо них в системе появятся новые уравнения вида $E' = E''$. После этого шага уравнения вида В) — это единственные нестандартные уравнения, которые еще остаются в системе.

6) Устранение нестандартных уравнений $E' = E''$. Решающее значение здесь имеет тот факт, что Δ -выражения E' и E'' являются префиксными. Это обусловлено тем, что исходная система уравнений $\mathcal{E}_1$ была построена для префиксного трансдьюсера π , и все преобразования, которые применяются в дальнейшем к этой системе, сохраняют, как это следует из утверждений 4–7, префиксное свойство Δ -выражений, входящих в уравнения. Для устранения нестандартного уравнения типа В)

$$\sum_{i=1}^{\ell} u_i \cdot X'_i = \sum_{j=1}^m v_j \cdot X''_j \quad (3)$$

проверяется совместность префиксных языков $L' = \{u_1, \dots, u_{\ell}\}$ и $L'' = \{v_1, \dots, v_m\}$. Если какое-нибудь слово одного из этих языков несовместно с любым словом другого языка, то процедура проверки завершает работу и объявляет о неразрешимости исходной системы уравнений на основании утверждения 14. В противном случае процедура осуществляет расщепление $L' = \bigcup_{i=1}^n L'_i$ и $L'' = \bigcup_{i=1}^n L''_i$ этих языков (см. утверждение 10), удаляет уравнение (3) из системы $\mathcal{E}_t$, и добавляет вместо него для каждой доли $L'_i = \{u_{i_0}\}$ и $L''_i = \{v_{i_1}, \dots, v_{i_r}\}$ (или $L'_i = \{u_{i_1}, \dots, u_{i_r}\}$ и $L''_i = \{v_{i_0}\}$) новое уравнение $X'_{i_0} = (v_{i_1} \setminus u_{i_0}) \cdot X''_{i_1} + \dots + (v_{i_r} \setminus u_{i_0}) \cdot X''_{i_r}$ (или, соответственно, $X'_{i_0} = (u_{i_1} \setminus v_{i_0}) \cdot X'_{i_1} + \dots + (u_{i_r} \setminus v_{i_0}) \cdot X'_{i_r}$). Утверждение 5 гарантирует при этом, что Δ -выражения в правых частях добавленных уравнений будут префиксными.

После этого шага будет получена система уравнений $\mathcal{E}_{t+1}$, которая равносильна системе $\mathcal{E}_t$, но имеет меньшее число активных переменных, чем $\mathcal{E}_t$.

Можно обратить внимание на тот факт, что описанная процедура проверки разрешимости систем уравнений очень похожа на алгоритм Мартелли–Монтанари [70] вычисления наиболее общего унификатора двух термов. И так же, как и в этом алгоритме унификации, правила 1)–6) преобразования систем уравнений в приведенную форму могут применяться в произвольном порядке.

Утверждение 16. Для любого префиксного трансдьюсера реального времени π и пары его состояний p, q предложенная процедура после применения к системе уравнений $\mathcal{E}_1 = \mathcal{E}_{\pi} \cup \{X_p = X_q\}$ завершает работу и корректно проверяет разрешимость системы $\mathcal{E}_1$.

Доказательство. Так как после каждого этапа t выполнения процедуры число активных переменных уменьшается по меньшей мере на единицу, процедура обязательно завершает работу. Преобразования по правилам 1) и 3)–6), которым подвергается система уравнений $\mathcal{E}_t$, являются равносильными преобразованиями; поэтому на каждом этапе t система уравнений $\mathcal{E}_t$ равносильна

исходной системе $\mathcal{E}_1$. Согласно утверждениям 13–15 решение относительно разрешимости системы уравнений $\mathcal{E}_t$, которое принимает процедура, является правильным. $\square$

Таким образом, верна

Теорема 2. *Проблема эквивалентности для префиксных трансдюсеров реального времени разрешима за квадратичное время.*

Доказательство. Согласно утверждению 16 описанная выше разрешающая процедура корректно проверяет эквивалентность $\pi(p) \sim \pi(q)$ для любого префиксного трансдюсера реального времени π . Число этапов работы этой процедуры не превосходит числа состояний трансдюсера π . Все Δ -выражения, входящие в систему уравнений $\mathcal{E}_t$, можно представить в виде словарного дерева с использованием ссылочных структур данных. Такие структуры данных позволяют выполнять все операции над уравнениями, такие как применение подстановки, проверка совместности, расщепление уравнений и др., за время, линейное относительно размера системы уравнений $\mathcal{E}_t$. Необходимо заметить, что использование ссылочных структур данных позволяет выполнять эти операции так, чтобы размер системы уравнений $\mathcal{E}_t$ не увеличивался. Поскольку исходную систему уравнений $\mathcal{E}_1$ можно построить за время, линейное относительно размера трансдюсера π , проверку эквивалентности $\pi(p) \sim \pi(q)$ можно выполнить за время, квадратичное относительно размера π . $\square$

6. Проверка эквивалентности префиксных трансдюсеров с ε -переходами

В этом разделе выделен новый класс префиксных трансдюсеров с ε -переходами и показано, как можно проверять эквивалентность машин из этого класса при помощи того же метода проверки разрешимости систем языковых уравнений, который был разработан в разделе 5. Но для трансдюсеров более общего вида коэффициентами этих уравнений будут не отдельные слова, а регулярные префиксные языки.

Для трансдюсера $\pi = \langle Q, F, \longrightarrow \rangle$, работающего в входном алфавите Σ и выходном алфавите Δ , назовем состояние q этого трансдюсера Σ -состоянием (ε -состоянием), если ни один ε -переход не исходит (соответственно, только ε -переходы исходят) из состояния q . Состояния обоих типов будем называть однородными. Трансдюсер π называется *однородным*, если все его состояния являются однородными. Каждый трансдюсер можно легко преобразовать в однородный: если из состояния q исходят переходы обоих типов, то его можно расщепить на два состояния q' и q'' так, чтобы q' унаследовало от q только исходящие из него Σ -переходы, а q'' унаследовало от q только исходящие из этого состояния ε -переходы. Переходы, которые прежде вели в состояние q , дублируются и направляются в оба состояния q' и q'' . Далее в этом разделе мы будем рассматривать только однородные трансдюсеры.

Множество состояний Q однородного трансдюсера можно разделить на подмножество Q_Σ , которое содержит все Σ -состояния, и подмножество Q_ε , которое содержит все ε -состояния. Точно так же, множество финальных состояний F разделяется на подмножество F_Σ , содержащее все финальные Σ -состояния, и подмножество F_ε , содержащее все финальные ε -состояния. Так как однородный трансдюсер может иметь циклы, состоящие только из ε -переходов, он способен, в отличие от трансдюсера реального времени, после прочтения одной буквы на входной ленте записать сколь угодно много различных слов на выходной ленте. Чтобы описать реакцию однородного трансдюсера π в ответ на событие начала вычисления или считывания буквы на входной ленте, мы вводим четыре класса языков:

- 1) для каждой пары Σ -состояний p, q и входной буквы $a \in \Sigma$ определим множество слов

$$Out_\pi^\Sigma(p, a, q) = \{ w : p \xrightarrow{a|w}^* q \},$$

которые π может записать на входной ленте при совершении прогона из состояния p в состояние q в результате считывания единственной буквы a ;

- 2) для каждого Σ -состояния p и входной буквы $a \in \Sigma$ определим множество слов

$$Fin_{\pi}^{\Sigma}(p, a) = \{w : p \xrightarrow{a|w}_{*} q, q \in F_{\Sigma}\},$$

которые π может записать на входной ленте при совершении прогона из состояния p в некоторое финальное ε -состояние в результате считывания единственной буквы a ;

- 3) для каждого ε -состояния p и Σ -состояния q определим множество слов

$$Out_{\pi}^{\varepsilon}(p, q) = \{w : p \xrightarrow{\varepsilon|w}_{*} q\},$$

которые π может записать на входной ленте при совершении некоторого ε -прогона из состояния p в состояние q ;

- 4) для каждого ε -состояния p определим множество слов

$$Fin_{\pi}^{\varepsilon}(p) = \{w : p \xrightarrow{\varepsilon|w}_{*} q, q \in F_{\varepsilon}\},$$

которые π может записать на входной ленте при совершении некоторого ε -прогона из состояния p в некоторое финальное ε -состояние.

Подобно тому, как это было определено для трансдюсеров реального времени, будем называть однородный трансдюсер π *префиксным*, если для любого $x \in \Sigma \cup \{\varepsilon\}$, состояния q , и пары различных переходов $q \xrightarrow{x|u'} q'$ и $q \xrightarrow{x|u''} q''$ выходные слова u' и u'' несовместны. Следующее утверждение вытекает непосредственно из приведенного определения.

Утверждение 17. Если $\pi = \langle Q_{\Sigma} \cup Q_{\varepsilon}, F_{\Sigma} \cup F_{\varepsilon}, \longrightarrow \rangle$ — это однородный префиксный трансдюсер, то

1. для каждого Σ -состояния p и входной буквы $a \in \Sigma$ все языки из семейства $\{Out_{\pi}^{\Sigma}(p, a, q) : q \in Q_{\Sigma}\}$ являются попарно вполне несовместными регулярными языками, которые не имеют расширений в регулярном языке $Fin_{\pi}^{\varepsilon}(p, a)$;
2. для каждого ε -состояния p все языки из семейства $\{Out_{\pi}^{\varepsilon}(p, q) : q \in Q_{\Sigma}\}$ являются попарно вполне несовместными регулярными языками, которые не имеют расширений в регулярном языке $Fin_{\pi}^{\varepsilon}(p)$.

Доказательство. Рассматривая каждый ε -переход $p \xrightarrow{\varepsilon|w} q$ трансдюсера π , где $w = x_1 \dots x_m$, как последовательную композицию $p \xrightarrow{x_1} \dots \xrightarrow{x_m} q$ переходов, можно построить для каждого ε -состояния p такой конечный мультиавтомат, который 1) работает в выходном алфавите Δ , 2) имеет p в качестве начального состояния, и 3) имеет в качестве семейства множеств допускающих состояний все одноэлементные множества $\{q\}, q \in Q_{\Sigma}$, а также множество F_{ε} . Так как трансдюсер π обладает свойством префиксности, указанный мультиавтомат является префиксным multi-DFA, который описывает семейство языков $\{Out_{\pi}^{\varepsilon}(p, q) : q \in Q_{\Sigma}\}$ и язык $Fin_{\pi}^{\varepsilon}(p)$.

Следуя тем же рассуждениям можно построить для каждого Σ -состояния p трансдюсера π и любой входной буквы $a \in \Sigma$ префиксный multi-DFA, который описывает семейство языков $\{Out_{\pi}^{\Sigma}(p, a, q) : q \in Q_{\Sigma}\}$ и язык $Fin_{\pi}^{\varepsilon}(p, a)$. $\square$

Заметим, что multi-DFAs, предложенные в доказательстве утверждения 17 можно построить за время, линейное относительно размера трансдюсера π , и размер каждого такого multi-DFA не превосходит размера машины π . Алгоритм проверки эквивалентности префиксных однородных трансдюсеров проводит анализ и преобразования семейств регулярных языков $Out_{\pi}^{\Sigma}(p, a, q)$, $Fin_{\pi}^{\Sigma}(p, a)$, $Out_{\pi}^{\varepsilon}(p, q)$, и $Fin_{\pi}^{\varepsilon}(p)$, используя указанные multi-DFAs как их формальные спецификации.

Вначале покажем, как построить систему уравнений $\mathcal{E}_\pi$, которая специфицирует отношения трансдукции, реализуемые в Σ -состояниях однородного трансдюсера $\pi = \langle Q_\varepsilon \cup Q_\Sigma, F_\varepsilon \cup F_\Sigma, \longrightarrow \rangle$. Уравнения состоят из r -выражений, которые, как и в случае трансдюсеров реального времени, строятся из переменных $X_1, X_2, \dots$, констант 0, 1, и букв алфавита Σ , но теперь вместо букв из алфавита Δ в качестве констант будут использоваться регулярные языки из класса $Reg(\Delta)$. Значением каждой такой константы $L \in Reg(\Delta)$ является трансдукция $\{(\varepsilon, w) : w \in L\}$. Разумеется, для этих новых констант будут верны тождества $L_1 + L_2 = L_3$ или $L_1 \cdot L_2 = L_3$ всякий раз, когда $L_1 \cup L_2 = L_3$ или $L_1 L_2 = L_3$. К каждой константе $L \in Reg(\Delta)$ будем обращаться по имени (описанию) какого-либо DFA, который задает язык L .

Обобщенное Δ -выражение — это всякое выражение вида $L_0 + L_1 \cdot X_1 + \dots + L_n \cdot X_n$, где $L_i \in Reg(\Delta)$ для каждого $i, 0 \leq i \leq n$. Обобщенное Δ -выражение такого вида называется префиксным, если языки $L_1, \dots, L_n$ являются префиксными, попарно вполне несовместными и не имеющими расширений в языке L_0 . Ясно, что каждое обобщенное Δ -выражение E может быть описано multi-DFA D_E , который задает семейство регулярных языков $L_0, L_1, \dots, L_n$. Обобщенное Σ -выражение — это всякое выражение вида $a_1 \cdot E_1 + \dots + a_k \cdot E_k + L$, где $\Sigma = \{a_1, \dots, a_k\}$, $L \in Reg(\Delta)$, и $E_1, \dots, E_k$ — это обобщенные Δ -выражения.

Для заданного однородного трансдюсера $\pi = \langle Q_\varepsilon \cup Q_\Sigma, F_\varepsilon \cup F_\Sigma, \longrightarrow \rangle$ определим систему уравнений $\mathcal{E}_\pi$ точно так же, как это было сделано для трансдюсеров реального времени. С каждым Σ -состоянием p будем ассоциировать переменную X_p , и для каждой входной буквы $a \in \Sigma$ построим обобщенное Δ -выражение $E_{p,a} = \sum_{q \in Q_\Sigma} Out_\pi^\Sigma(p, a, q) \cdot X_q + Fin_\pi^\Sigma(p, a)$. Нетрудно заметить, что если π — префиксный трансдюсер, то всякое обобщенное Δ -выражение $E_{p,a}$ является префиксным. Поведение трансдюсера π можно описать следующей системой уравнений

$$\mathcal{E}_\pi = \{X_p = \sum_{a \in \Sigma} a \cdot E_{p,a} + L_p : p \in Q_\Sigma\},$$

в которых $L_p = 1$, если $p \in F_\Sigma$, и $L_p = 0$ в противном случае.

Утверждение 18. Для каждого однородного трансдюсера π система уравнений $\mathcal{E}_\pi$ имеет единственное решение $\{X_p = TR(\pi, p) : p \in Q_\Sigma\}$.

Доказательство. Проводится посредством тех же самых рассуждений, что и доказательство аналогичного утверждения 12. $\square$

Для каждого состояния $p \in Q_\Sigma \cup Q_\varepsilon$ однородного трансдюсера π определим обобщенное Δ -выражение $E_{\pi,p}$:

$$E_{\pi,p} = \begin{cases} X_p, & \text{если } p \in Q_\Sigma, \\ \sum_{q \in Q_\Sigma} Out_\pi^\varepsilon(p, q) X_q + Fin_\pi^\varepsilon(p), & \text{если } p \in Q_\varepsilon. \end{cases}$$

Утверждение 19. Для каждого однородного трансдюсера π и пары его состояний $p', p'' \in Q_\Sigma \cup Q_\varepsilon$ система уравнений $\mathcal{E}_\pi \cup \{E_{\pi,p'} = E_{\pi,p''}\}$ имеет решение тогда и только тогда, когда $\pi(p') \sim \pi(p'')$.

Доказательство. Согласно утверждению 18 система уравнений $\mathcal{E}_\pi$ имеет единственное решение $X_p = TR(\pi, p)$ для каждого Σ -состояния p . Тогда, как видно из определения множеств $Out_\pi^\varepsilon(p, q)$ и $Fin_\pi^\varepsilon(p)$, для этого набора значений переменных Δ -выражения $E_{\pi,p'}$ и $E_{\pi,p''}$ принимают значения $TR(\pi, p')$ и $TR(\pi, p'')$. $\square$

Таким образом, проблема эквивалентности $\pi(p') \stackrel{?}{\sim} \pi(p'')$ для однородных трансдюсеров сводится к задаче проверки разрешимости систем уравнений $\mathcal{E}_\pi \cup \{E_{\pi,p'} = E_{\pi,p''}\}$. И хотя в общем случае эта задача не имеет алгоритмов решения, для нее существует эффективная разрешающая процедура, если известно, что π — префиксный трансдюсер.

Далее будет представлена процедура проверки разрешимости систем уравнений вида $\mathcal{E}_\pi \cup \{E'_1 = E''_1, \dots, E'_m = E''_m\}$, где π — префиксный однородный трансдюсер, и E'_i, E''_i — префиксные Δ -выражения для всех $1 \leq i \leq m$. Как и в случае трансдюсеров реального времени, процедура основана на методе Гаусса исключения переменных и заключается в преобразовании исходной системы $\mathcal{E}_1$ к равносильной системе уравнений, представленной в приведенной форме

$$\mathcal{E}_{\pi'}(X_1, \dots, X_n) \cup \{Y_j = E_j(X_1, \dots, X_n) : 1 \leq j \leq r\},$$

путем поочередной деактивизации переменных системы $\mathcal{E}_1$.

Уравнения каждой такой системы $\mathcal{E}_t = \mathcal{E}_\pi \cup \{E'_1 = E''_1, \dots, E'_m = E''_m\}$ подразделяются, как и в случае трансдюсеров реального времени, на базовые, приведенные и активные. На каждой итерации к системе уравнений $\mathcal{E}_t$ применяются следующие преобразования.

- 1) *Удаление тождеств.* Все активные уравнения $E' = E''$, являющиеся тождествами, удаляются из системы $\mathcal{E}_t$. Для этой цели достаточно проверить эквивалентность multi-DFAs $D_{E'}$ и $D_{E''}$, которые описывают Δ -выражения E' и E'' .
- 2) *Проверка активности переменных.* Если $\mathcal{E}_t$ не содержит активных уравнений, то процедура завершает свое выполнение и объявляет о разрешимости системы уравнений в соответствии с утверждением 18.
- 3) *Обработка активных уравнений.* Из системы $\mathcal{E}_t$ выбирается какое-нибудь активное уравнение

$$\sum_{i=1}^n L'_i X_i + L'_0 = \sum_{i=1}^n L''_i X_i + L''_0. \quad (4)$$

Если $L'_i = L''_i$ для всех $i, 1 \leq i \leq n$, то процедура завершает свое выполнение и объявляет о неразрешимости системы уравнений. Основания для этого решения таковы. Так как уравнение (4) не является тождеством, L'_0 и L''_0 — это различные языки. Поскольку языки $L'_i, 1 \leq i \leq n$, не имеют расширений в обоих языках L'_0 и L''_0 , для любых трансдукций $T_1, \dots, T_n$ верно, что

$$L'_0 \cap \bigcup_{i=1}^n L'_i T_i = L''_0 \cap \bigcup_{i=1}^n L''_i T_i = \emptyset.$$

Поэтому левая и правая части уравнения (4) всегда будут иметь разные значения.

Если $L'_j \neq L''_j$ для некоторого $j, 1 \leq j \leq n$, то выберем самое короткое свидетельство того, что эти языки не равны, т.е. кратчайшее выходное слово u в непустом языке $\bigcup_{i=1}^n (L'_i \oplus L''_i)$. Будем далее полагать для определенности, что $u \in L'_1$ и при этом $u \notin L''_1$. Так как Δ -выражение $\sum_{i=1}^n L'_i X_i + L'_0$ является префиксным, все языки $L'_2, \dots, L'_n$ не имеют расширений в u . Поэтому, принимая во внимание тот факт, что u — это кратчайшее слово в множестве слов $\bigcup_{i=1}^n (L'_i \oplus L''_i)$, можно заключить, что языки $L''_1, \dots, L''_n$ также не имеют расширений в u . Поэтому, ввиду того, что u не имеет расширений в языке L'_0 , для любых трансдукций $T_1, \dots, T_n$ верно, что

$$\sum_{i=1}^n L'_i T_i + L'_0 = \sum_{i=1}^n L''_i T_i + L''_0 \Rightarrow T_1 = \sum_{i=1}^n (L''_i \setminus u) T_i + (L''_0 \setminus u).$$

Таким образом, $\mathcal{E}_t$ равносильна системе уравнений $\mathcal{E}_t \cup \{X_1 = \sum_{i=1}^n (L_i'' \setminus u)X_i + (L_0'' \setminus u)\}$.

Рассмотрим это дополнительное уравнение

$$X_1 = \sum_{i=1}^n (L_i'' \setminus u)X_i + (L_0'' \setminus u). \quad (5)$$

Если $L_0'' \setminus u = L_1'' \setminus u = \dots = L_n'' \setminus u = \emptyset$, то процедура завершает свое выполнение и объявляет о неразрешимости системы уравнений $\mathcal{E}_t$, так как согласно утверждению 18, подсистема $\mathcal{E}_\pi$ имеет своими решениями только непустые трансдукции.

В противном случае, необходимо заметить, что соотношение $\varepsilon \notin L_1'' \setminus u$ имеет место, ввиду указанной ранее особенности выбора слова u . Поэтому на основании теоремы Ардена [67, 71], уравнение (5) равносильно уравнению

$$X_1 = \sum_{i=2}^n (L_1'' \setminus u)^*(L_i'' \setminus u)X_i + (L_1'' \setminus u)^*(L_0'' \setminus u). \quad (6)$$

Таким образом, уравнение (6) может быть введено в систему $\mathcal{E}_t$ вместо уравнения (5), и после этого все вхождения переменной X_1 в других уравнениях системы $\mathcal{E}_t$ можно заменить на Δ -выражение из правой части уравнения (6). В результате будет получена равносильная система уравнений, в которой появляется новое приведенное уравнение (6) и, тем самым, число приведенных переменных увеличивается на 1.

Но применение указанной подстановки к уравнениям системы $\mathcal{E}_t$ имеет те же побочные эффекты, что и в случае с трансдюсерами реального времени: возникают нестандартные уравнения вида

А) $E = G$, где E — это отличное от переменной Δ -выражение, а G — это Σ -выражение, и

В) $E' = E''$, где E', E'' — это отличные от переменных Δ -выражения,

и, кроме того,

С) несколько базовых уравнений вида $X = G$ с одной и той же переменной X в их левых частях могут появиться $\mathcal{E}_t$. Тогда каноническую форму системы уравнений можно восстановить при помощи равносильных преобразований 4) и 5), описанных в разделе 5. Утверждения 2–9 гарантируют, что после применения этих преобразований (включая и применение подстановки) все Δ -выражения в полученной системе уравнений $\mathcal{E}_{t+1}$ остаются префиксными.

Утверждение 20. Для каждого префиксного однородного трансдюсера π и пары его состояний p', p'' описанная процедура после применения к системе уравнений $\mathcal{E}_1 = \mathcal{E}_\pi \cup \{E_{\pi, p'} = E_{\pi, p''}\}$ всегда завершает работу и корректно устанавливает разрешимость системы $\mathcal{E}_1$.

Доказательство. На каждой итерации t процедура либо завершает свою работу и корректно обнаруживает (не)разрешимость системы уравнений $\mathcal{E}_t$, либо строит равносильную систему уравнений $\mathcal{E}_{t+1}$, которая имеет большее количество разрешенных переменных, чем система $\mathcal{E}_t$. В результате процедура проверки либо обнаруживает, что система уравнений $\mathcal{E}_1$ не имеет решений, либо строит равносильную приведенную систему уравнений, которая согласно утверждению 18, свидетельствует о разрешимости исходной системы уравнений $\mathcal{E}_1$. $\square$

Теорема 3. Проблема эквивалентности для префиксных однородных трансдюсеров разрешима за кубическое время.

Доказательство. Разрешимость проблемы эквивалентности следует из утверждений 19 и 20. Число итераций, которые должна совершить описанная в этом разделе процедура проверки разрешимости, не превосходит числа состояний анализируемого трансдюсера π . Как можно видеть из определения языков $Out_\pi^\Sigma(p, a, q)$, $Out_\pi^\varepsilon(p)$, $Fin_\pi^\Sigma(p, a)$, $Fin_\pi^\varepsilon(p)$, multi-DFAs, которые описывают Δ -выражения

в системе уравнений $\mathcal{E}_\pi$ и верифицирующем уравнении $E_{\pi,p'} = E_{\pi,p''}$ могут быть извлечены без всяких вычислительных издержек из системы переходов проверяемого трансдюсера π . В разделе 3 было показано, что все преобразования Δ -выражений, которые осуществляет процедура проверки разрешимости, можно выполнить за линейное время без увеличения общего размера используемых multi-DFAs. И только две операции — проверка равенства регулярных языков $L(A') = L(A'')$, представленных автоматами, и вычисление кратчайшего слова, свидетельствующего о том, что эти языки не равны, т.е. слова $u \in L(A') \oplus L(A'')$ из симметричной разности двух регулярных языков, описанных при помощи DFAs A' и A'' , — требуют больших вычислительных затрат. Обе эти операции можно выполнить за время, зависящее квадратично от размеров участвующих в этих операциях DFAs (см. [67, 72]). $\square$

В следующих разделах будет показано, как можно воспользоваться префиксными однородными трансдюсерами для проверки эквивалентности детерминированных двухленточных автоматов и детерминированных биавтоматов.

7. Проверка эквивалентности детерминированных двухленточных автоматов

Двухленточный конечный автомат (далее, 2-FSA), работающий в алфавитах Σ и Δ , задается системой переходов $M = \langle S_1, S_2, F, \rightarrow \rangle$; эта система определяется подмножествами состояний S_1, S_2 , на которые разбито конечное множество состояний $S = S_1 \cup S_2$, подмножеством *финальных состояний* $F \subseteq S$, и *отношением переходов* $\rightarrow$, которое имеет тип $(S_1 \times \Sigma \times S) \cup (S_2 \times \Delta \times S)$. Переходы будем обозначать записями $s \xrightarrow{z} s'$, где $z \in \Sigma$ в случае $s \in S_1$, и $z \in \Delta$ в случае $s \in S_2$. Условимся использовать обозначение $\rightarrow_\Delta$ для проекции отношения переходов $\rightarrow$ на алфавит Δ , т.е. $\rightarrow_\Delta = \rightarrow \cap S_2 \times \Delta \times S$.

Прогон 2-FSA M — это любая последовательность переходов вида

$$s \xrightarrow{z_1} s_1 \xrightarrow{z_2} \dots \xrightarrow{z_{n-1}} s_{n-1} \xrightarrow{z_n} s' . \quad (7)$$

Прогон (7) называется *финальным*, если $s' \in F$. Говорят, что 2-FSA M *допускает* пару слов $(h, w) \in \Sigma^* \times \Delta^*$, начиная работу в состоянии s , если у автомата M есть такой финальный прогон (7), что h — это проекция слова $z_1 z_2 \dots z_{n-1} z_n$ на алфавит Σ , а w — это проекция того же самого слова $z_1 z_2 \dots z_{n-1} z_n$ на алфавит Δ . *Трансдукция, распознаваемая 2-FSA M из состояния s* — это множество $TR(M, s)$ всех пар слов (h, w) , допускаемых машиной M , которая начинает работу из указанного состояния. 2-FSA M называется *детерминированным* (2-DFSA), если для каждой буквы z и состояния s он имеет не более одного перехода вида $s \xrightarrow{z} s'$.

Состояния s' и s'' из множества состояний 2-FSAs M считаются *эквивалентными* (для обозначения этого отношения используем запись $M(s') \sim M(s'')$), если $TR(M, s') = TR(M, s'')$. *Проблема эквивалентности* для 2-FSAs состоит в том, чтобы для заданного 2-FSA M и заданной пары его состояний s' и s'' выяснить выполнимость отношения $M(s') \sim M(s'')$.

Взаимосвязь между 2-FSAs и трансдюсерами очевидна: если потребовать, чтобы двухленточный автомат вместо считывания букв на второй ленте, записывал те же самые буквы этой ленты, то в результате 2-FSA превращается в однородный трансдюсер, который вычисляет в своих состояниях те же самые отношения трансдукции. Если система переходов $M = \langle S_1, S_2, F, \rightarrow \rangle$ не имеет циклов, проходящих только по состояниям множества S_2 (т.е. автомат не может сколь угодно долго считывать буквы на второй ленте), то 2-FSA M можно превратить в эквивалентный трансдюсер реального времени. Однако оставалось неясным, какому классу трансдюсеров соответствуют 2-DFSA. Выделив класс префиксных однородных трансдюсеров, мы можем дать ответ на этот вопрос.

Рассмотрим произвольный 2-FSA $M = \langle S_1, S_2, F, \dashrightarrow \rangle$, и построим соответствующий ему трансдьюсер $\pi_M = \langle S_1 \cup S_2, F, \longrightarrow \rangle$, определив для любой пары состояний s, s' и всякой буквы $x \in \Sigma \cup \Delta$ отношение переходов по следующим правилам:

- π_M имеет переход $s \xrightarrow{x|\varepsilon} s'$ тогда и только тогда, когда $s \in S_1$, $x \in \Sigma$, и M имеет переход $s \dashrightarrow^x s'$,
- π_M имеет переход $s \xrightarrow{\varepsilon|y} s'$ тогда и только тогда, когда $s \in S_2$, $y \in \Delta$, и M имеет переход $s \dashrightarrow^y s'$.

Утверждение 21. Для произвольного заданного 2-DFSA M соответствующий ему трансдьюсер π_M является префиксным и однородным, и при этом равенство $TR(M, s) = TR(\pi_M, s)$ выполняется для любого состояния $s \in S_1 \cup S_2$.

Доказательство. Как показывает определение отношения переходов трансдьюсера π_M , из любого его состояния s существует финальный прогон, помеченный парой слов (h, w) , в том и только том случае, когда 2-DFSA M , начиная работу в состоянии s , допускает эту пару слов. Поэтому $TR(M, s) = TR(\pi_M, s)$.

Что касается свойства префиксности, то для его подтверждения достаточно заметить, что если $S_1 = \{s_1, \dots, s_m\}$, то для каждого состояния $s \in S_2$ multi-DFA $D = \langle S, s, \{s_1\}, \dots, \{s_m\}, F \cap S_2, \dashrightarrow_\Delta \rangle$ является префиксным. $\square$

Объединив утверждение 21 и теорему 3, получаем решение проблемы эквивалентности для 2-DFSA.

Теорема 4. Проблема эквивалентности для детерминированных двухленточных конечных автоматов разрешима за время, кубическое относительно их размеров.

8. Проблема эквивалентности для детерминированных биавтоматов

Конечный биавтомат (BA), работающий в алфавите Σ , задается системой переходов $B = \langle Q, F, Left, Right \rangle$, в которой Q — это конечное множество состояний, $F \subseteq Q$ — это подмножество финальных состояний, а $Left$ и $Right$ — это отношения переходов типа $Q \times \Sigma \times Q$.

Биавтомат B имеет две головки, которые считывают на ленте буквы входного слова. Левая головка считывает буквы с левого конца слова, а правая головка — с правого конца этого же слова. На каждом шаге вычисления биавтомат B выбирает одну из головок, считывает с ее помощью очередную букву входного слова и переходит в новое состояние в соответствии с отношением переходов $Left$ (если для считывания была выбрана левая головка) или $Right$ (если букву считывала правая головка). Вычисление завершается, как только головки встречаются на ленте друг с другом после прочтения всех букв входного слова. Если по окончании вычисления биавтомат B оказывается в финальном состоянии, то он допускает входное слово.

Вычисления биавтомата удобнее всего определить в терминах конфигураций. Конфигурацией биавтомата B называется пара $\alpha = (q, w)$, в которой q — это состояние B , а w — это слово в алфавите Σ (та часть входного слова, которую еще предстоит прочитать биавтомату). Отношение $\vdash$ на множестве конфигураций биавтомата B определяется для каждой пары его состояний q, q' , слова w , и буквы a следующим образом: $(q, aw) \vdash (q', w)$ выполняется, если $(q, a, q') \in Left$, и $(q, wa) \vdash (q', w)$ выполняется, если $(q, a, q') \in Right$. Прогоном биавтомата B на входном слове w из состояния q называется всякая последовательность конфигураций

$$\alpha_0 \vdash \alpha_1 \vdash \dots \alpha_{n-1} \vdash \alpha_n, \quad (8)$$

в которой $\alpha_0 = (q, w)$. Прогон (8) называется финальным в том случае, когда $\alpha_n = (q', \varepsilon)$ для некоторого состояния $q' \in F$. Биавтомат B допускает входное слово w из состояния q , если существует его финальный прогон на слове w из состояния q . Язык, распознаваемый биавтоматом B из состояния

q , — это множество $L(B, q)$ всех слов, которые биавтомат B допускает из этого состояния. Условимся использовать запись $\mathcal{L}(BA)$ для обозначения класса языков, распознаваемых биавтоматами.

Состояния q' и q'' биавтомата B считаются *эквивалентными* (и этот факт обозначается записью $B(q') \sim B(q'')$), если $L(B, q') = L(B, q'')$. Проблема эквивалентности для биавтоматов состоит в том, чтобы для произвольного заданного биавтомата B и произвольной заданной пары его состояния q' и q'' выяснить, выполняется ли отношение $B(q') \sim B(q'')$.

В статьях [12, 14, 43] были выделены некоторые специальные классы биавтоматов. Биавтомат $B = \langle Q, F, Left, Right \rangle$ называется *детерминированным* (DBA), если множество его состояний Q разбито на два подмножества Q_L и Q_R так, что отношения переходов $Left$ и $Right$ представляют собой (частичные) функции $Left : Q_L \times \Sigma \rightarrow Q$ и $Right : Q_R \times \Sigma \rightarrow Q$. DBA B называется *ослабленным справа* (DWBA), если $F \subseteq Q_L$ и для любого состояния $q \in Q_R$ существует единственная буква $a \in \Sigma$, для которой определено значение функции $Right(q, a)$. Луканова в статье [12] доказала, что $\mathcal{L}(BA)$ совпадает с классом **Lin** линейных к-с языков, и поэтому проблема эквивалентности для произвольных биавтоматов неразрешима. Йираскова и Клима установили в статье [14], что $\mathcal{L}(DWBA)$ совпадает с классом **DLin** детерминированных линейных к-с языков, и это означает, что проблема эквивалентности для DWBAs разрешима. Однако в работе [43] было показано, что $\mathcal{L}(DBA)$ несравнимо с семейством детерминированных к-с языков; это означает, что метод проверки эквивалентности DPDAs, разработанный в статье [60], неприменим для решения проблемы эквивалентности для DBAs, и поэтому вопрос об алгоритмической разрешимости этой задачи оставался открытым.

Мы покажем, как можно эффективно проверять эквивалентность DBAs, используя взаимосвязь между DBAs и префиксными трансдюсерами и применяя метод исключения переменных для проверки разрешимости систем линейных уравнений.

В статье [1] Розенберг обнаружил простое соответствие между 2-FSAs и линейными к-с языками: для любого линейного к-с языка L существуют такие 2-FSA M и его состояние s , для которых выполняется равенство $L = \{uv^{-1} : (u, v) \in TR(M, s)\}$. Так как биавтоматы распознают линейные к-с языки, а 2-FSAs — бинарные отношения, вычислимые однородными трансдюсерами, появляется возможность транслировать биавтоматы в трансдюсеры.

Для заданного биавтомата $B = \langle Q, F, Left, Right \rangle$, работающего в алфавите Σ , рассмотрим трансдюсер $\pi_B = \langle Q, F, \rightarrow \rangle$, который имеет одно и то же множество букв Σ в качестве входного и выходного алфавитов и отношение переходов которого определено следующим образом: для каждой пары состояний p, q и буквы a четверка $p \xrightarrow{a|a} q$ (или $p \xrightarrow{\varepsilon|a} q$) является переходом трансдюсера π_B в том и только том случае, если $(p, a, q) \in Left$ (или $(p, a, q) \in Right$ соответственно).

Утверждение 22. Для любого биавтомата $B = \langle Q, F, Left, Right \rangle$, соответствующего ему трансдюсера π_B и состояния $q \in Q$ верно, что

1. $L(B, q) = \{uv^{-1} : (u, v) \in TR(\pi_B, q)\}$;
2. Если биавтомат B детерминированный, то π_B — это префиксный однородный трансдюсер.

Доказательство. Следует непосредственно из определений DBA, языка $L(B, q)$, распознаваемого биавтоматом BA B , отношения трансдукции $TR(\pi, q)$, вычисляемого произвольным трансдюсером π , и указанного выше соответствия между биавтоматом B и трансдюсером π_B . $\square$

Будем говорить, что трансдюсер π порождает язык сцепления $LC(\pi, q) = \{uv^{-1} : (u, v) \in TR(\pi, q)\}$ из состояния q . Состояния p и q трансдюсера π считаются *ctn-эквивалентными* (обозначается $\pi(p) \sim_{ctn} \pi(q)$), если $LC(\pi, p) = LC(\pi, q)$. Таким образом, согласно утверждению 22, проблема эквивалентности для DBAs сводится к проблеме ctn-эквивалентности для префиксных однородных трансдюсеров.

Для проверки ctn -эквивалентности $\pi(p) \sim_{\text{ctn}} \pi(q)$ префиксных однородных трансдюсеров можно применить тот же метод, при помощи которого в разделе 6 осуществлялась проверка эквивалентности $\pi(p) \sim \pi(q)$: построить систему языковых уравнений $\hat{\mathcal{E}}_\pi$, описывающую языки сцепления, порождаемые трансдюсером π , добавить к этой системе проверочное уравнение $X_p = X_q$ и проверить разрешимость полученной системы уравнений методом Гаусса исключения переменных. Но на этот раз регулярные выражения, из которых формируются уравнения, будут интерпретироваться в полукольце языков в алфавите Σ .

Уравнения системы $\hat{\mathcal{E}}_\pi$ строятся из r -выражений, а сами выражения образуются из переменных $X_1, X_2, \dots$, констант 0, 1, и регулярных языков из семейства $\text{Reg}(\Sigma)$. Регулярные языки расцениваются как константы, их именами служат те DFAs, которые описывают эти языки. Условимся считать $\hat{\Delta}$ -выражением $\hat{E}$ всякое выражение вида $L_0 + L_1 \cdot X_1 + \dots + L_n \cdot X_n$, где $L_i \in \text{Reg}(\Sigma)$ для каждого $i, 0 \leq i \leq n$. И хотя здесь рассматриваются трансдюсеры, у которых входные и выходные слова задаются в одном и том же алфавите Σ , в целях сохранения единообразия введенная ранее классификация r -выражений будет сохранена. Будем называть $\hat{\Delta}$ -выражение $\hat{E}$ префиксным, если языки $L_1, \dots, L_n$ являются префиксными, попарно вполне несравнимыми и свободными от расширения в языке L_0 . Для представления любого $\hat{\Delta}$ -выражения такого рода можно использовать multi-DFA $D_{\hat{E}}$, который описывает семейство регулярных языков $L_1, \dots, L_n, L_0$. Будем называть $\hat{\Sigma}$ -выражением всякое выражение $\hat{G}$ вида $\hat{E}_1 \cdot a_1 + \dots + \hat{E}_k \cdot a_k + L$, где $\Sigma = \{a_1, \dots, a_k\}$, $L \in \text{Reg}(\Sigma)$, и $\hat{E}_1, \dots, \hat{E}_k$ — это $\hat{\Delta}$ -выражения.

Для произвольного заданного однородного трансдюсера $\pi = \langle Q_\varepsilon \cup Q_\Sigma, F_\varepsilon \cup F_\Sigma, \longrightarrow \rangle$ система уравнений $\hat{\mathcal{E}}_\pi$ определяется так. Для каждого состояния $p \in Q$ и буквы $a \in \Sigma$ конструируется $\hat{\Delta}$ -выражение $\hat{E}_{p,a} = \sum_{q \in Q_\Sigma} \text{Out}_\pi^\Sigma(p, a, q) \cdot X_q + \text{Fin}_\pi^\Sigma(p, a)$. Нетрудно заметить, что если π — это префиксный трансдюсер, то каждое $\hat{\Delta}$ -выражение $\hat{E}_{p,a}$ также является префиксным. Семейство языков сцепления, порождаемое трансдюсером π задается следующей системой уравнений

$$\hat{\mathcal{E}}_\pi = \{X_p = \sum_{a \in \Sigma} \hat{E}_{p,a} \cdot a + L_p : p \in Q_\Sigma\},$$

в которой $L_p = 1$, если $p \in F_\Sigma$, и $L_p = 0$ в противном случае. Следуя той же схеме доказательства, что и для утверждения 12 и принимая во внимание изменения, которые были внесены в определение r -выражений, можно легко доказать

Утверждение 23. Для каждого однородного трансдюсера π система уравнений $\hat{\mathcal{E}}_\pi$ имеет единственное решение $\{X_p = LC^{-1}(\pi, p) : p \in Q_\Sigma\}$.

Чтобы записать проверочные уравнения, определим для каждого состояния p однородного трансдюсера π специальное $\hat{\Delta}$ -выражение $\hat{E}_{\pi,p}$:

$$\hat{E}_{\pi,p} = \begin{cases} X_p, & \text{если } p \in Q_\Sigma, \\ \sum_{q \in Q_\Sigma} \text{Out}_\pi^\varepsilon(p, q) \cdot X_q + \text{Fin}_\pi^\varepsilon(p), & \text{если } p \in Q_\varepsilon. \end{cases}$$

Утверждение 24. Для каждого однородного трансдюсера π и любой пары состояний $p', p'' \in Q_\Sigma \cup Q_\varepsilon$ система уравнений $\hat{\mathcal{E}}_\pi \cup \{\hat{E}_{\pi,p'} = \hat{E}_{\pi,p''}\}$ имеет решение тогда и только тогда, когда $\pi(p') \sim_{\text{ctn}} \pi(p'')$.

Доказательство. Как следует из утверждения 23, система уравнений $\hat{\mathcal{E}}_\pi$ имеет единственное решение $X_p = LC^{-1}(\pi, p)$ для каждого состояния $p \in Q_\Sigma$. Обратившись к определениям множеств $\text{Out}_\pi^\varepsilon(p, q)$ и $\text{Fin}_\pi^\varepsilon(p)$, можно заметить, что для такого решения $\hat{\Delta}$ -выражения $\hat{E}_{\pi,p'}$ и $\hat{E}_{\pi,p''}$ в обеих частях проверочного уравнения принимают значения $LC^{-1}(\pi, p')$ и $LC^{-1}(\pi, p'')$. $\square$

Разрешимость систем уравнений $\hat{\mathcal{E}}_\pi \cup \{\hat{E}'_1 = \hat{E}''_1, \dots, \hat{E}'_m = \hat{E}''_m\}$, где π — это префиксный однородный трансдюсер, и $\hat{E}'_i, \hat{E}''_i, 1 \leq i \leq m$, — это префиксные $\hat{\Delta}$ -выражения, можно проверять, придерживаясь того же самого сценария по деактивизации переменных, который был описан в разделе 6, т.е. преобразуя системы уравнений к равносильной приведенной форме. На каждой итерации процедура проверки применяет следующие равносильные преобразования к сложившейся системе уравнений $\hat{\mathcal{E}}_t$.

- 1) Удаление тождеств $\hat{E} = \hat{E}$.
- 2) Проверка активности переменных.
- 3) Обработка активных уравнений $\hat{E}' = \hat{E}''$.
- 4) Удаление нестандартных уравнений вида $\hat{E} = \hat{G}$.

Эти преобразования совершенно аналогичны тем правилам, которые описаны в разделе 6. Единственное отличие имеет правило удаления нестандартных уравнений вида $\hat{G}' = \hat{G}''$, где $\hat{G}', \hat{G}''$ — Σ -выражения.

- 5) Удаление нестандартных уравнений вида $\hat{G}' = \hat{G}''$.

Если система содержит уравнение

$$\sum_{i=1}^k \hat{E}'_i \cdot a_i + L' = \sum_{i=1}^k \hat{E}''_i \cdot a_i + L'',$$

то нужно удалить его из системы и добавить вместо него более простые новые уравнения $\hat{E}'_i + L'/a_i = \hat{E}''_i + L''/a_i, 1 \leq i \leq k$. Нетрудно заметить, что в результате применения этого правила будет получена равносильная система уравнений. Согласно утверждению 3, в новых уравнениях $\hat{\Delta}$ -выражения по обе стороны равенства остаются префиксными. Таким образом, все нестандартные уравнения вида $\hat{G}' = \hat{G}''$ исчезают из системы $\hat{\mathcal{E}}_t$, и мы получаем равносильную систему уравнений $\hat{\mathcal{E}}_{t+1}$ с меньшим числом активных переменных. Как и в разделе 6, для представления и обработки $\hat{\Delta}$ -выражений используются multi-DFAs. Указанные соображения, вкупе с утверждениями 23 и 24 приводят к следующему результату.

Теорема 5. *Проблема эквивалентности для детерминированных конечных биавтоматов разрешима за кубическое время.*

Описанный здесь метод проверки эквивалентности применим только к детерминированным биавтоматам (DBAs). Решающая роль в нем отводится правилам 3) деактивизации переменных и 5) удаления нестандартных уравнений вида $\hat{G}' = \hat{G}''$. Эти правила требуют, чтобы трансдюсер π_B обладал свойством префиксности, а оно является следствием детерминированности биавтомата B .

Разрешающий алгоритм можно существенно упростить для некоторых классов DBAs. Например, если биавтомат не имеет возможности сколь угодно долго считывать буквы, оставаясь при этом все время на правой стороне слова, то его можно транслировать в префиксный трансдюсер реального времени, для которого проверку ctn-эквивалентности состояний можно проводить значительно эффективнее, чем для префиксных однородных трансдюсеров.

Для произвольного DBA $B = \langle Q_L, Q_R, F, Left, Right \rangle$ его функцию переходов $Right : Q_R \times \Sigma \rightarrow Q_L \cup Q_R$ можно обычным образом распространить на слова из Σ^* : для любых состояния $q \in Q_R$, слова $w \in \Sigma^*$ и буквы $x \in \Sigma$ будем полагать $Right^*(q, \varepsilon) = q$ и $Right^*(q, wx) = Right(p, x)$, если $Right^*(q, w) = p \in Q_R$. DBA B назовем *детерминированным ациклическим справа биавтоматом* (DABA), если $F \subseteq Q_L$ и $Right^*(q, w) \neq q$ для любого состояния $q \in Q_R$ и всякого непустого слова $w \neq \varepsilon$. Как видно из этого определения, каждый детерминированный ослабленный справа биавтомат (DWBA) является ациклическим справа. При этом верны строгие включения $\mathcal{L}(DWBA) \subset \mathcal{L}(DABA) \subset \mathcal{L}(DBA)$, поскольку

- язык $\{a^n b^n : n \geq 0\} \cup \{a^n c^n : n \geq 0\}$, очевидно, принадлежит классу $\mathcal{L}(DABA)$, но не принадлежит семейству языков $\mathcal{L}(DWBA)$;
- язык $\{a^n (bc^*)^n : n \geq 0\} : n \geq 0\}$, очевидно, принадлежит классу $\mathcal{L}(DBA)$, но не принадлежит семейству языков $\mathcal{L}(DABA)$.

Сопоставим $DBA B = \langle Q_L, Q_R, F, Left, Right \rangle$ трансдюсер $\pi_B = \langle Q_L, F, \longrightarrow \rangle$, в котором отношение переходов $\longrightarrow$ определено следующим образом: для любой пары состояний $p, q \in Q_L$, буквы $x \in \Sigma$ и слова $w \in \Sigma^*$ трансдюсер π_B имеет переход $p \xrightarrow{x|w} q$ в том и только том случае, если выполнено одно из двух условий

1. $Left(p, x) = q$ и $w = \varepsilon$, или
2. $Left(p, x) = r \in Q_R$ и $Right^*(r, w) = q$.

Для предложенной альтернативной трансляции биавтоматов в трансдюсеры реального времени верно

Утверждение 25. Для любого $DABA B$ соответствующий ему трансдюсер π_B является префиксным конечным трансдюсером реального времени, и для любого его состояния $q \in Q_L$ верно равенство $L(B, q) = \{uv^{-1} : (u, v) \in TR(\pi_B, q)\}$.

Доказательство. Так как биавтомат является ациклическим справа, для любого состояния $r \in Q_R$ обобщенная функция переходов $Right^*(r, w)$ определена на конечном множестве слов w . Поэтому, как следует из определения отношения переходов для трансдюсера π_B , для всякой буквы $x \in \Sigma$ из любого состояния $p \in Q_L$ исходит конечное число переходов вида $p \xrightarrow{x|w} q$. Поскольку биавтомат B является детерминированным, для любого состояния $q \in Q_R$ множество слов $\{w : Right^*(q, w) \in Q_L\}$ обладает свойством префиксности. Поэтому, как следует из определения отношения переходов для трансдюсера π_B , для любого состояния p и всякой буквы x множества слов $Out_{\pi_B}(p, q, x)$, где $q \in Q_L$, являются попарно вполне несовместными префиксными языками. Значит, π_B — префиксный трансдюсер реального времени. И, наконец, как это было уже отмечено в утверждении 22, равенство $L(B, q) = \{uv^{-1} : (u, v) \in TR(\pi_B, q)\}$ непосредственно следует из определения языка $L(B, q)$ и отношения переходов для π_B . $\square$

Таким образом, проблема эквивалентности для $DABAs$ может быть сведена к задаче проверки ctn -эквивалентности состояний префиксных конечных трансдюсеров реального времени, и эту задачу можно решить путем проверки разрешимости соответствующих систем линейных уравнений так же, как это было показано ранее для префиксных однородных трансдюсеров. Отличие будет состоять в том, что теперь коэффициентами Δ -выражений будут не префиксные регулярные языки, а отдельные слова. Поэтому для проверки разрешимости таких систем уравнений можно воспользоваться модифицированной процедурой, описанной в разделе 5 и основанной на утверждении 10. Эта модификация, по существу, касается только формы уравнений и приводит к следующим результатам.

Теорема 6. Проблема эквивалентности для детерминированных ациклических справа конечных биавтоматов ($DABAs$) разрешима за квадратичное время.

Следствие 2. Проблема эквивалентности для детерминированных ослабленных справа конечных биавтоматов ($DWBAs$) разрешима за квадратичное время.

Следствие 3. Проблема эквивалентности для детерминированных линейных k -с грамматик разрешима за квадратичное время.

9. Проблема эквивалентности для простых грамматик

Метод Гаусса исключения переменных можно также успешно применять для проверки разрешимости систем нелинейных уравнений, которые описывают языки, порождаемые некоторыми к-с грамматиками, и, таким образом, получать эффективное решение проблемы эквивалентности для этих грамматик. Впервые алгебраическим подходом к решению проблемы эквивалентности для к-с грамматик воспользовались авторы статьи [45], доказав разрешимость указанной задачи для т.н. простых грамматик. В этом разделе мы покажем, как можно подходящим образом приспособить общую схему метода исключения переменных для проверки разрешимости систем уравнений, описывающих задачу проверки эквивалентности нетерминалов простых грамматик.

Контекстно-свободная грамматика в алфавите Σ задается парой $\Gamma = \langle \mathcal{N}, \mathcal{P} \rangle$, состоящей из конечного множества *нетерминалов* $\mathcal{N}$ и конечного множества *грамматических правил* (продукций) $\mathcal{P}$. Символы множества $\mathcal{N}$ (нетерминалы) отличны от букв алфавита Σ (терминалов). Для удобства именования условимся последовательности букв в алфавите Σ называть терминальными словами и обозначать символами u, v, w , а последовательности букв в алфавите $\Sigma \cup \mathcal{N}$ называть предложениями и обозначать символами α, β, γ . Грамматические правила — это пары вида $N \rightarrow \alpha$, где N — нетерминал из множества $\mathcal{N}$, который называется *заголовком правила*, а α — предложение, которое называется *телом правила*. Грамматика Γ задает *отношение непосредственной выводимости* $\rightarrow_\Gamma$ на множестве предложений: два предложения β_1, β_2 находятся в отношении непосредственной выводимости $\beta_1 \rightarrow_\Gamma \beta_2$ в том и только том случае, если $\beta_1 = \beta' N \beta''$, $\beta_2 = \beta' \alpha \beta''$ и при этом множество $\mathcal{P}$ содержит правило $N \rightarrow \alpha$. *Отношение грамматического вывода* $\xrightarrow{*}_\Gamma$, порождаемое грамматикой Γ , — это рефлексивно-транзитивное замыкание отношения непосредственной выводимости $\rightarrow_\Gamma$. *Язык предложения* α в грамматике Γ — это множество терминальных слов $L(\Gamma, \alpha) = \{ w : w \in \Sigma^*, \alpha \xrightarrow{*}_\Gamma w \}$, выводимых из предложения α в грамматике Γ .

Нетерминал N называется *несущественным* в грамматике Γ , если $L(\Gamma, N) = \emptyset$. Как показано в учебнике [67], все несущественные нетерминалы к-с грамматики можно выявить за квадратичное время. Все правила, содержащие несущественные нетерминалы, можно удалить из грамматики Γ , и в результате будет получена грамматика Γ' , для любого нетерминала которой верно равенство $L(\Gamma, N) = L(\Gamma', N)$. Поэтому при изучении проблемы эквивалентности ограничиваются рассмотрением к-с грамматик, не содержащих несущественных нетерминалов.

Два предложения α_1, α_2 считаются *эквивалентными в грамматике* Γ (обозначается $\alpha_1 \sim_\Gamma \alpha_2$), если $L(\Gamma, \alpha_1) = L(\Gamma, \alpha_2)$. Нетрудно заметить, что для к-с грамматик отношение эквивалентности предложений является отношением конгруэнтности относительно операции конкатенации: соотношение $\beta \sim_\Gamma \gamma \Rightarrow \alpha' \beta \alpha'' \sim_\Gamma \alpha' \gamma \alpha''$ выполняется для любых предложений $\alpha', \alpha'', \beta, \gamma$.

Проблема эквивалентности нетерминалов состоит в том, чтобы для любой заданной грамматики Γ и для любой пары ее нетерминалов N_1, N_2 проверить выполнимость отношения эквивалентности $N_1 \sim_\Gamma N_2$.

Проблему эквивалентности нетерминалов будем решать для класса простых грамматик. Грамматическое правило находится в *нормальной форме*, если оно имеет вид $N \rightarrow x\alpha$, где x — терминал, который называется *ключом правила*, а $\alpha \in \mathcal{N}^*$ — цепочка нетерминалов. К-с грамматика Γ называется *простой* [45], если все правила грамматики находятся в нормальной форме, и при этом все правила с одинаковым заголовком имеют попарно различные ключи.

Как видно из определения, простая грамматика Γ обладает свойством однозначности: для любого нетерминала N и слова $w \in L(\Gamma, N)$ существует единственный левосторонний грамматический вывод w из N . Это свойство проявляет себя в следующем утверждении, которое важно для решения проблемы эквивалентности.

Утверждение 26. Пусть $\Gamma = \langle \mathcal{N}, \mathcal{P} \rangle$ — простая грамматика. Тогда для любых предложений $\alpha_1, \alpha_2, \beta_1, \beta_2 \in \mathcal{N}^*$ и терминального слова w верно соотношение

$$\alpha_1 \sim_{\Gamma} \alpha_2 \wedge \alpha_1 \xrightarrow{*}_{\Gamma} w\beta_1 \wedge \alpha_2 \xrightarrow{*}_{\Gamma} w\beta_2 \Rightarrow \beta_1 \sim_{\Gamma} \beta_2.$$

В статье [45] было установлено, что для любой простой грамматики $\Gamma = \langle \mathcal{N}, \mathcal{P} \rangle$ и нетерминала $N \in \mathcal{N}$ язык $L(\Gamma, N)$ является префиксным. Поэтому, как заметили авторы указанной статьи, для решения задач анализа простых грамматик целесообразно опираться на некоторые основополагающие свойства префиксных языков. К их числу относятся законы левого и правого сокращения префиксных языков относительно операции конкатенации.

Утверждение 27 ([45]). Для любых префиксных языков L, L', L'' выполняются следующие соотношения:

- 1) $LL' = LL'' \iff L' = L''$,
- 2) $L'L = L''L \iff L' = L''$.

Алгоритм проверки эквивалентности двух нетерминалов M, N произвольной заданной простой грамматики $\Gamma = \langle \mathcal{N}, \mathcal{P} \rangle$ состоит из двух этапов: построение системы уравнений $\mathcal{E}(\Gamma, M, N)$, которая описывает проблему эквивалентности, и проверка разрешимости построенной системы уравнений.

Уравнения системы $\mathcal{E}(\Gamma, M, N)$ строятся из выражений, а сами выражения образуются из переменных $X_1, X_2, \dots$ и констант при помощи операций конкатенации $\cdot$ и альтернации $+$. Каждый нетерминал X рассматривается как переменная. Константами являются $0, 1$, и терминалы из алфавита Σ . Все выражения интерпретируются в полукольце языков в алфавите Σ .

Для каждого нетерминала X формируется базовое уравнение $X = E_X$, где $E_X = \sum_{X \rightarrow \alpha \in \mathcal{P}} \alpha$. Система уравнений, описывающая задачу проверки эквивалентности $M \stackrel{?}{\sim}_{\Gamma} N$, имеет вид $\mathcal{E}(\Gamma, M, N) = \mathcal{E}_{\Gamma} \cup \{M = N\}$, где $\mathcal{E}_{\Gamma} = \{X = E_X : X \in \mathcal{N}\}$.

Утверждение 28 ([45]). Для любой простой грамматики Γ , не содержащей несущественных нетерминалов, система базовых уравнений $\mathcal{E}_{\Gamma}$ имеет единственное решение $X = L(\Gamma, X)$ для каждого нетерминала X .

Следствие 4. Для любой простой грамматики Γ , не содержащей несущественных нетерминалов, система уравнений $\mathcal{E}(\Gamma, M, N) = \mathcal{E}_{\Gamma} \cup \{M = N\}$ имеет решение тогда и только тогда, когда $M \sim_{\Gamma} N$.

На втором этапе алгоритм проверки эквивалентности $M \sim_{\Gamma} N$ применяет процедуру проверки разрешимости систем уравнений вида

$$\mathcal{E}_t = \mathcal{E}_{\Gamma} \cup \{\alpha_1 = \beta_1, \dots, \alpha_m = \beta_m\}, \quad (9)$$

где $\alpha_i, \beta_i \in \mathcal{N}^*$. Проверочное уравнение $\alpha_i = \beta_i$ системы (9) называется *приведенным*, если предложение α_i состоит из единственной переменной X_j , которая не имеет других вхождений в уравнения системы; переменная X_j в этом случае также называется *приведенной*. Система уравнений (9) называется *приведенной*, если все проверочные уравнения являются приведенными. Как следует из утверждения 28, всякая приведенная система уравнений имеет решение. На основании этого, процедура проверки разрешимости систем уравнений вида (9) применяет равносильные преобразования в стремлении построить приведенную систему. На каждой итерации эта процедура либо увеличивает число приведенных переменных, либо обнаруживает несовместные уравнения, и это означает, что система не имеет решения.

Процедура начинает работу с системой уравнений $\mathcal{E}_1 = \mathcal{E}(\Gamma, M, N)$, и на каждой итерации t применяет к сложившейся системе уравнений (9) следующие правила.

1) *Упрощение уравнений.* К обеим частям проверочных уравнений применяются законы левого и правого сокращения (утверждение 27) $X\alpha = X\beta \Rightarrow \alpha = \beta$ и $\alpha X = \beta X \Rightarrow \alpha = \beta$. Если в результате применения этих преобразований образуется тождество $1 = 1$, то оно удаляется из системы. В том случае если одно из проверочных уравнений преобразуется к виду $X\alpha = 1$ или $1 = X\beta$, то процедура проверки прекращает работу и объявляет систему уравнений $\mathcal{E}(\Gamma, M, N)$ неразрешимой. Если таких несовместных уравнений не возникнет, то после сокращений переменных все оставшиеся в системе проверочные уравнения будут иметь вид $X\alpha = Y\beta$, где X и Y — различные переменные.

2) *Проверка активности.* Если в системе $\mathcal{E}_t$ все проверочные уравнения $\alpha_i = \beta_i$ являются приведенными, то $\mathcal{E}_t$ — приведенная система уравнений. Тогда процедура проверки разрешимости завершает свое выполнение и объявляет о разрешимости исходной системы уравнений $\mathcal{E}(\Gamma, M, N)$.

3) *Построение уравнений вида $X = \alpha$.* Если в системе $\mathcal{E}_t$ содержится проверочное уравнение $X\alpha = Y\beta$, и при этом $|\alpha| > 0$ и $|\beta| > 0$, то проводится сравнение длин кратчайших терминальных слов u и v , выводимых из нетерминалов X и Y : $X \xrightarrow{*}_{\Gamma} u$, $Y \xrightarrow{*}_{\Gamma} v$. Далее будем полагать, что $|u| \leq |v|$ (в противном случае левую и правую части уравнения можно поменять местами). Если нельзя построить левосторонний грамматический вывод вида $Y \xrightarrow{*}_{\Gamma} u\gamma$, то процедура проверки прекращает работу и объявляет систему уравнений $\mathcal{E}(\Gamma, M, N)$ неразрешимой. А если $Y \xrightarrow{*}_{\Gamma} u\gamma$ для некоторого предложения $\gamma \in \mathcal{N}^*$, то уравнение $X\alpha = Y\beta$ в системе $\mathcal{E}_t$ заменяется парой уравнений $\alpha = \gamma\beta$ и $Y = X\gamma$. После этого в системе $\mathcal{E}_t$ появляется хотя бы одно проверочное уравнение вида $Y = X\gamma$, левая часть которого состоит только из одной переменной и отлична от правой части.

4) *Приведение переменных.* В получившейся системе $\mathcal{E}_t$ выбирается неприведенное уравнение вида $Y = \beta$, в котором предложение β отлично от переменной Y . Если Y содержится в β , то процедура проверки прекращает работу и объявляет систему уравнений $\mathcal{E}(\Gamma, M, N)$ неразрешимой. Если предложение β не содержит переменной Y , то ко всем остальным уравнениям системы применяется подстановка $\{Y/\beta\}$. После ее применения проверочное уравнение $Y = \beta$ становится приведенным, и, таким образом, число приведенных уравнений системы увеличивается.

Однако применение подстановки $\{Y/\beta\}$ к базовым уравнениям системы имеет побочный эффект: в системе могут возникнуть либо два базовых уравнения с одинаковой левой частью, либо уравнение вида $X\beta = \sum_{a \in \Sigma} a \cdot \alpha_a$, где $|\beta| > 0$. Поэтому далее необходимо применить преобразование, которое восстанавливает канонический вид системы.

5) *Устранение дубликатов.* Если в системе возникают два уравнения вида $X = \sum_{a \in \Sigma} a \cdot \alpha_a$ и $X\beta = \sum_{a \in \Sigma} a \cdot \alpha'_a$, левые части которых начинаются одинаковой переменной, то второе уравнение удаляется из системы, и вместо него в систему добавляются проверочные уравнения $\alpha_a\beta = \alpha'_a$, $a \in \Sigma$. В результате применения этого преобразования канонический вид получившейся системы уравнений будет восстановлен, и процедура проверки разрешимости переходит к следующей итерации $t + 1$.

Утверждение 29. Для любой простой грамматики Γ и пары ее нетерминалов M, N описанная процедура после применения к системе уравнений $\mathcal{E}(\Gamma, M, N)$ всегда завершает работу и корректно устанавливает разрешимость этой системы.

Доказательство. Описанная процедура всегда завершает работу, т.к. после каждой итерации t число разрешенных переменных системы $\mathcal{E}_{t+1}$ увеличивается.

Правила проверки разрешимости задают равносильные преобразования систем уравнений. Для правил 1), 4) и 5) это следует из утверждения 27 и конгруэнтности отношения $\sim_{\Gamma}$. Рассмотрим правило 3) и допустим, что $X \xrightarrow{*}_{\Gamma} u$ и $Y \xrightarrow{*}_{\Gamma} u\gamma$. Тогда $X\alpha \xrightarrow{*}_{\Gamma} u\alpha$ и $Y\beta \xrightarrow{*}_{\Gamma} u\gamma\beta$, и, значит, согласно утверждениям 26 и 27 из равенства $X\alpha = Y\beta$ следуют равенства $\alpha = \gamma\beta$ и $X\gamma = Y$. Таким образом, если $X \xrightarrow{*}_{\Gamma} u$ и $Y \xrightarrow{*}_{\Gamma} u\gamma$, то уравнение $X\alpha = Y\beta$ равносильно системе из двух уравнений

$\alpha = \gamma\beta$ и $Y = X\gamma$. Значит, правило 3) также задает равносильное преобразование. Поэтому на каждой итерации t процедура работает с системой уравнений $\mathcal{E}_t$, которая равносильна исходной системе $\mathcal{E}(\Gamma, M, N)$.

Процедура правильно проверяет разрешимость исходной системы уравнений. Мы показали, что каждая система уравнений $\mathcal{E}_t$ равносильна системе $\mathcal{E}_\Gamma \cup \{M = N\}$. Согласно утверждению 28 единственным решением системы уравнений $\mathcal{E}_\Gamma$ являются значения переменных-нетерминалов $X = L(\Gamma, X)$. Заметим, что правила вывода простых грамматик устроены так, что $\varepsilon \notin L(\Gamma, X)$ для любого нетерминала X . Поэтому значения $X = L(\Gamma, X)$ не удовлетворяют уравнениям вида $X\alpha = 1$ или $X = \beta'X\beta''$. Очевидно также, что если эти значения переменных удовлетворяют равенству $X\alpha = Y\beta$, то для любого слова $w \in L(\Gamma, X)$ язык $L(\Gamma, Y)$ содержит либо префикс, либо расширение слова w . Поэтому правила 1), 3) и 4) правильно устанавливают неразрешимость системы $\mathcal{E}_t$. Так как всякая приведенная система уравнений имеет решение, правило 2) правильно распознает разрешимость системы $\mathcal{E}_t$. $\square$

Для простой грамматики Γ обозначим записью $\nu(\Gamma)$ величину $\max_{N \in \mathcal{N}} \min_{w \in L(\Gamma, N)} |w|$. Из утверждения 29 вытекает

Теорема 7. *Проблема эквивалентности нетерминалов для произвольной простой грамматики Γ разрешима за время $O(n^3 \nu(\Gamma))$, где n — суммарный размер правил грамматики Γ .*

Доказательство. Разрешимость проблемы эквивалентности следует из утверждения 29. Для представления уравнений используются ссылочные структуры данных. Процедура, проверяющая разрешимость системы уравнений, совершает не более n итераций и на каждой из них, имея систему проверочных уравнений размера m , может выполнить правила 1) и 4) за время $O(m)$, правила 2) и 5) — за время $O(n)$, правило 3) — за время $O(n\nu(\Gamma))$, и увеличить при этом размер системы проверочных уравнений на величину $O(n\nu(\Gamma))$. Отсюда следует указанная оценка сложности. $\square$

Заключение

По мнению автора этой статьи, потенциальные возможности предложенного алгебраического подхода к построению быстрых алгоритмов проверки эквивалентности различных видов автоматов не исчерпываются теми результатами, которые были здесь представлены. Можно отметить несколько направлений, по которым целесообразно продолжить исследования, связанные с дальнейшим развитием разработанного метода и расширением его области применения.

Во-первых, вполне естественно попробовать применить алгебраический метод проверки эквивалентности к другим моделям вычислений, поведение которых можно описать при помощи систем алгебраических уравнений. Главной целью здесь можно наметить задачу построения полиномиального по времени алгоритма проверки эквивалентности детерминированных k -ленточных автоматов для $k \geq 3$. Однако свойства префиксных языков, благоприятствующие построению быстрого алгоритма проверки эквивалентности детерминированных двухленточных автоматов и проявляющиеся, например, в утверждении 10, уже не смогут, по-видимому, помочь решению этой более трудной задачи. Необходимо, насколько это возможно, отыскать принципиально новые приемы восстановления канонического вида системы уравнений после применения подстановок для разрешенных переменных (см. правила 4)-6) в разделе 5). Вполне вероятно, что достичь этого удастся за счет исследования проблемы эквивалентности для некоторых сравнительно узких классов многоленточных автоматов, которые можно транслировать в детерминированные конечные автоматы-преобразователи, работающие над специальными полугруппами (см. [34]), в которых возможно, проявляются свойства, полезные для эффективной проверки разрешимости систем уравнений, аналогичные указанным выше свойствам префиксных языков.

Другое направление развития алгебраического метода проверки эквивалентности автоматов — это построение с его помощью алгоритмов минимизации различных типов автоматов, и, в первую очередь, детерминированных автоматов-преобразователей (трансдюсеров), работающих над специальными полугруппами. Успешный пример построения эффективных алгоритмов минимизации трансдюсеров на основе алгоритмов проверки эквивалентности представлен в статьях [8, 73]. Одной из интересных задач для дальнейших исследований может быть задача разработки полиномиальных по времени алгоритмов минимизации детерминированных двухленточных автоматов или биавтоматов. Здесь можно также опираться на результаты статьи [10].

И, наконец, заслуживает внимания задача построения эффективных алгоритмов проверки эквивалентности и минимизации в классе конечных автоматов-преобразователей, работающих над полугруппами подстановок. Как показано в статье [9], именно эта автоматная модель вычислений в наибольшей мере соответствует модели последовательных императивных программ, семантика которых задается отношением логико-термальной эквивалентности [74]. Таким образом, на этом пути исследований может быть обнаружена возможность непосредственного использования алгебраического метода проверки эквивалентности автоматов для решения задач оптимизации компьютерных программ.

References

- [1] A. L. Rosenberg, “A machine realization of the linear context-free languages”, *Information and Control*, vol. 10, no. 2, pp. 175–188, 1967.
- [2] M. Mohri, “Finite-state transducers in language and speech processing”, *Computational linguistics*, vol. 23, no. 2, pp. 269–311, 1997.
- [3] A. Roche-Lima and R. K. Thulasiram, “Bioinformatics algorithm based on a parallel implementation of a machine learning approach using transducers”, *Journal of Physics: Conference Series*, vol. 341, no. 1, p. 012 034, 2012.
- [4] R. Alur and P. Černý, “Streaming transducers for algorithmic verification of single-pass list-processing programs”, in *Proceedings of 38-th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*, 2011, pp. 599–610.
- [5] J. Thakkar and R. Kanade A. and Alur, “Transducer-based algorithmic verification of retransmission protocols over noisy channels”, in *Proceedings of IFIP Joint International Conference on Formal Techniques for Distributed Systems*, Springer, 2013, pp. 209–224.
- [6] M. Veanes, P. Hooimeijer, B. Livshits, and et al., “Symbolic finite state transducers: Algorithms and applications”, in *Proceedings of the 39-th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*, 2012, pp. 137–150.
- [7] D. G. Kozlova and V. A. Zakharov, “On the model checking of sequential reactive systems”, in *Proceedings of the 25-th International Workshop on Concurrency, Specification and Programming*, Rostock, Germany, September 28-30, vol. 1698, 2016, p. 233.
- [8] M. Mohri, “Minimization algorithms for sequential transducers”, *Theoretical Computer Science*, vol. 234, no. 2, pp. 177–201, 2000.
- [9] V. A. Zakharov and S. R. Jaylauova, “On the minimization problem for sequential programs”, *Automatic Control and Computer Sciences*, vol. 51, no. 7, pp. 689–700, 2017.
- [10] H. Tamm, M. Nykänen, and E. Ukkonen, “On size reduction techniques for multitape automata”, *Theoretical computer science*, vol. 363, no. 2, pp. 234–246, 2006.
- [11] D. C. Luckham, D. M. Park, and M. S. Paterson, “On formalized computer programs”, *Journal of Computer and System Sciences*, vol. 4, no. 3, pp. 220–249, 1970.

- [12] R. Loukanova, “Linear context free languages”, in *International Colloquium on Theoretical Aspects of Computing*, Springer, 2007, pp. 351–365.
- [13] B. Bedregal, “ λ -ALN: autômatos lineares não-determinísticos com λ -transições”, *Tendências em Matemática Aplicada e Computacional*, vol. 12, no. 3, pp. 171–182, 2011.
- [14] G. Jiraskova and O. Klima, “Deterministic Biautomata and Subclasses of Deterministic Linear Languages”, in *International Conference on Language and Automata Theory and Applications*, Springer, 2019, pp. 315–327.
- [15] M. Holzer and S. Jakobi, “Nondeterministic biautomata and their descriptive complexity”, in *International Workshop on Descriptive Complexity of Formal Systems*, Springer, 2013, pp. 112–123.
- [16] O. Klima and L. Polak, “On biautomata*”, *RAIRO-Theoretical Informatics and Applications*, vol. 46, no. 4, pp. 573–592, 2012.
- [17] M. Holzer and S. Jakobi, “Minimization and characterizations for biautomata”, *Fundamenta Informaticae*, vol. 136, no. 1-2, pp. 113–137, 2015.
- [18] P. C. Fischer and A. L. Rosenberg, “Multitape one-way nonwriting automata”, *Journal of Computer and System Sciences*, vol. 2, no. 1, pp. 88–101, 1968.
- [19] T. Griffiths, “The unsolvability of the equivalence problem for Λ -free nondeterministic generalized machines”, *Journal of the ACM (JACM)*, vol. 15, no. 3, pp. 409–413, 1968.
- [20] O. Ibarra, “The unsolvability of the equivalence problem for ε -free NGSMS with unary input (output) alphabet and applications”, *SIAM Journal on Computing*, vol. 7, no. 4, pp. 524–532, 1978.
- [21] M. Blattner and T. Head, “Single-valued a-transducers”, *Journal of Computer and System Sciences*, vol. 15, no. 3, pp. 310–327, 1977.
- [22] M. P. Schützenberger, “Sur les relations rationnelles”, in *Proceedings of the Conference on Automata Theory and Formal Languages*, 1975, pp. 209–213.
- [23] M. Blattner and T. Head, “The decidability of equivalence for deterministic finite transducers”, *Journal of Computer and System Sciences*, vol. 19, no. 1, pp. 45–49, 1979.
- [24] E. M. Gurari and O. Ibarra, “A note on finite-valued and finitely ambiguous transducers”, *Mathematical systems theory*, vol. 16, no. 1, pp. 61–66, 1983.
- [25] A. Weber, “On the valuedness of finite transducers”, *Acta Informatica*, vol. 27, no. 8, pp. 749–780, 1990.
- [26] K. Culik and J. Karhumäki, “The equivalence of finite valued transducers (on HDTOL languages) is decidable”, *Theoretical Computer Science*, vol. 47, pp. 71–84, 1986.
- [27] A. Weber, “A decomposition theorem for finite-valued transducers and an application to the equivalence problem”, in *Proceedings of the 13-th International Symposium on Mathematical Foundations of Computer Science*, Springer, 1988, pp. 552–562.
- [28] A. Weber, “Decomposing finite-valued transducers and deciding their equivalence”, *SIAM Journal on Computing*, vol. 22, no. 2, pp. 175–202, 1993.
- [29] M.-P. Béal, O. Carton, C. Prieur, and J. Sakarovitch, “Squaring transducers: an efficient procedure for deciding functionality and sequentiality”, *Theoretical Computer Science*, vol. 292, no. 1, pp. 45–63, 2003.
- [30] J. Sakarovitch and R. de Souza, “On the decomposition of k -valued rational relations”, in *Proceedings of the 25-th International Symposium on Mathematical Foundations of Computer Science*, 2008, pp. 588–600.
- [31] J. Sakarovitch and R. de Souza, “On the decidability of bounded valuedness for transducers”, in *Proceedings of the 25-th International Symposium on Mathematical Foundations of Computer Science*, 2008, pp. 588–600.

- [32] J. Sakarovitch and R. de Souza, “Lexicographic decomposition of k -valued transducers”, *Theory of Computing Systems*, vol. 47, no. 3, pp. 758–785, 2010.
- [33] R. de Souza, “On the decidability of the equivalence for k -valued transducers”, in *Proceedings of the 12-th International Conference on Developments in Language Theory*, Springer, 2008, pp. 252–263.
- [34] V. A. Zakharov, “Equivalence checking problem for finite state transducers over semigroups”, in *Proceedings of the 6-th International Conference on Algebraic Informatics*, Springer, 2015, pp. 208–221.
- [35] A. Weber, “On the lengths of values in a finite transducer”, *Acta Informatica*, vol. 29, no. 6-7, pp. 663–687, 1992.
- [36] R. de Souza, “On the Decidability of the Equivalence for a Certain Class of Transducers”, in *Proceedings of the 13-th International Conference on Developments in Language Theory*, Springer, 2009, pp. 478–489.
- [37] M. Bird, “The equivalence problem for deterministic two-tape automata”, *Journal of Computer and System Sciences*, vol. 7, no. 2, pp. 218–236, 1973.
- [38] L. G. Valiant, “The equivalence problem for deterministic finite-turn pushdown automata”, *Information and Control*, vol. 25, no. 2, pp. 123–133, 1974.
- [39] C. Beeri, “An improvement on Valiant’s decision procedure for equivalence of deterministic finite turn pushdown machines”, *Theoretical Computer Science*, vol. 3, no. 3, pp. 305–320, 1976.
- [40] E. P. Friedman and S. A. Greibach, “A polynomial time algorithm for deciding the equivalence problem for 2-tape deterministic finite state acceptors”, *SIAM Journal on Computing*, vol. 11, no. 1, pp. 166–183, 1982.
- [41] T. Harju and J. Karhumäki, “The equivalence problem of multitape finite automata”, *Theoretical Computer Science*, vol. 78, no. 2, pp. 347–355, 1991.
- [42] J. Worrell, “Revisiting the equivalence problem for finite multitape automata”, in *Proceedings of the 40-th International Colloquium on Automata, Languages, and Programming*, Springer, 2013, pp. 422–433.
- [43] B. Bedregal, “Nondeterministic Linear Automata and a Class of Deterministic Linear Languages”, *Preliminary Proceedings LSFA*, pp. 183–196, 2015.
- [44] Y. Bar-Hillel, M. Perles, and E. Shamir, “On formal properties of simple phrase structure grammars”, *Sprachtypologie und Universalienforschung*, vol. 14, pp. 143–172, 1961.
- [45] A. J. Korenjak and J. E. Hopcroft, “Simple deterministic languages”, in *7th Annual Symposium on Switching and Automata Theory (swat 1966)*, IEEE, 1966, pp. 36–46.
- [46] E. P. Friedman, “The inclusion problem for simple languages”, *Theoretical Computer Science*, vol. 1, no. 4, pp. 297–316, 1976.
- [47] D. Caucal, “A fast algorithm to decide on simple grammars equivalence”, in *International Symposium on Optimal Algorithms*, Springer, 1989, pp. 66–85.
- [48] C. Bastien, J. Czyzowicz, W. Fraczak, and W. Rytter, “Prime normal form and equivalence of simple grammars”, in *International Conference on Implementation and Application of Automata*, Springer, 2005, pp. 78–89.
- [49] Y. Hirshfeld, M. Jerrum, and F. Moller, “A polynomial algorithm for deciding bisimilarity of normed context-free processes”, *Theoretical Computer Science*, vol. 158, no. 1-2, pp. 143–159, 1996.
- [50] L. G. Valiant and M. S. Paterson, “Deterministic one-counter automata”, *Journal of Computer and System Sciences*, vol. 10, no. 3, pp. 340–350, 1975.

- [51] S. Böhm and S. Göller, “Language equivalence of deterministic real-time one-counter automata is NL-complete”, in *International Symposium on Mathematical Foundations of Computer Science*, Springer, 2011, pp. 194–205.
- [52] G. Sénizergues, “The equivalence problem for t-turn dpda is co-NP”, in *Proceedings of the 30-th International Colloquium on Automata, Languages, and Programming*, Springer, 2003, pp. 478–489.
- [53] M. Linna, “Two decidability results for deterministic pushdown automata”, *Journal of Computer and System Sciences*, vol. 18, no. 1, pp. 92–107, 1979.
- [54] V. Y. Meytus, “The equivalence problem for real-time strict deterministic pushdown automata”, *Cybernetics*, vol. 25, no. 2, pp. 581–594, 1989.
- [55] M. Oyamaguchi, “The equivalence problem for real-time DPDAs”, *Journal of the ACM*, vol. 34, no. 3, pp. 731–760, 1987.
- [56] V. Y. Romanovskii, “Equivalence problem for real-time deterministic pushdown automata”, *Cybernetics*, vol. 22, no. 2, pp. 162–175, 1985.
- [57] D. J. Rosenkrantz and R. E. Stearns, “Properties of deterministic top-down grammars”, *Information and Control*, vol. 17, no. 3, pp. 226–256, 1970.
- [58] E. Tomita, “An extended direct branching algorithm for checking equivalence of deterministic pushdown automata”, *Theoretical Computer Science*, vol. 32, no. 1-2, pp. 87–120, 1984.
- [59] E. Ukkonen, “The equivalence problem for some non-real-time deterministic pushdown automata”, *Journal of the ACM*, vol. 29, no. 4, pp. 1166–1181, 1982.
- [60] G. Sénizergues, “The equivalence problem for deterministic pushdown automata is decidable”, in *Proceedings of the 24-th International Colloquium on Automata, Languages, and Programming*, Springer, 1997, pp. 671–681.
- [61] C. Stirling, “Deciding DPDA equivalence is primitive recursive”, in *Proceedings of the 29-th International Colloquium on Automata, Languages, and Programming*, Springer, 2002, pp. 821–832.
- [62] R. Madhavan, M. Mayer, S. Gulwani, and V. Kuncak, “Automating grammar comparison”, in *Proceedings of the 2015 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications*, 2015, pp. 183–200.
- [63] V. A. Zakharov, “Program equivalence checking by two-tape automata”, *Cybernetics and Systems Analysis*, vol. 46, no. 4, pp. 554–562, 2010.
- [64] T. Olshansky and A. Pnueli, “A direct algorithm for checking equivalence of LL(k) grammars”, *Theoretical Computer Science*, vol. 4, no. 3, pp. 321–349, 1977.
- [65] J. Karhumäki, “Equations over finite sets of words and equivalence problems in automata theory”, *Theoretical computer science*, vol. 108, no. 1, pp. 103–118, 1993.
- [66] A. Okhotin, “Decision problems for language equations”, *Journal of Computer and System Sciences*, vol. 76, no. 3-4, pp. 251–266, 2010.
- [67] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation (3rd ed.)* Addison-Wesley, 2006.
- [68] S. Eilenberg, *Automata, languages, and machines*. Academic press, 1974.
- [69] Y. Han, K. Salomaa, and D. Wood, “State Complexity of Prefix-Free Regular Languages.”, in *DCFS*, 2006, pp. 165–176.
- [70] A. Martelli and U. Montanari, “An efficient unification algorithm”, *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 2, pp. 258–282, 1982.

- [71] D. Arden, "Delayed-logic and finite-state machines", in *Proceedings of 2-nd Annual Symposium on Switching Circuit Theory and Logical Design (SWCT 1961)*, IEEE, 1961, pp. 133–151.
- [72] J. E. Hopcroft, *A linear algorithm for testing equivalence of finite automata*. Defense Technical Information Center, 1971, vol. 114.
- [73] V. A. Zakharov and G. G. Temerbekova, "On the minimization of finite state transducers over semigroups", *Automatic Control and Computer Sciences*, vol. 51, no. 7, pp. 523–530, 2017.
- [74] V. E. Itkin, "Logiko-termalnaya ekvivalentnost skhem programm", *Kibernetika i sistemnyj analiz*, no. 1, pp. 5–27, 1972.

On the Existence Problem of Finite Bases of Identities in the Algebras of Recursive Functions

V. A. Sokolov¹

DOI: [10.18255/1818-1015-2020-3-304-315](https://doi.org/10.18255/1818-1015-2020-3-304-315)

¹P. G. Demidov Yaroslavl State University, 14 Sovetskaya, Yaroslavl 150003, Russia.

MSC2020: 03D20

Research article

Full text in Russian

Received August 21, 2020

After revision September 7, 2020

Accepted September 9, 2020

Raphael Robinson showed that all primitive recursive functions depending on one argument, and only they could be obtained from two functions $s(x) = x + 1$ and $q(x) = x - [\sqrt{x}]^2$ by using operations of addition $+$, superposition $*$ and iteration i . Julia Robinson proved that from the same two functions, using the addition $+$, superposition $*$ and operation $^{-1}$ of function inversion, one could obtain all general recursive functions (under a certain condition on the inversion operation) and all partially recursive functions. On the basis of these results, A. I. Maltsev brought into consideration the Raphael Robinson algebra of all unary primitive recursive functions and two Julia Robinson algebras: the partial algebra of all unary general recursive functions and the algebra of all unary partially recursive functions and proposed to study the properties of these algebras, including the question of the existence of finite bases of identities in these algebras. In this article we show that there is no finite basis of identities in any of the indicated algebras.

Keywords: algebras; recursive functions; identities; basis; superposition; iteration; function inversion.

INFORMATION ABOUT THE AUTHORS

Valery Anatolyevich Sokolov
correspondence author

orcid.org/0000-0003-1427-4937. E-mail: valery-sokolov@yandex.ru

Doctor of Science, Professor, Department of Theoretical Informatics, Centre of Integrable Systems.

Funding: The initiative program VIP-004 (state registration number AAAA-A16-116070610022-6).

For citation: V. A. Sokolov, "On the Existence Problem of Finite Bases of Identities in the Algebras of Recursive Functions", *Modeling and analysis of information systems*, vol. 27, no. 3, pp. 304-315, 2020.

О проблеме существования конечных базисов тождеств в алгебрах рекурсивных функций

В. А. Соколов¹

DOI: [10.18255/1818-1015-2020-3-304-315](https://doi.org/10.18255/1818-1015-2020-3-304-315)

¹Ярославский государственный университет им. П. Г. Демидова, ул. Советская, 14, г. Ярославль, 150003 Россия.

УДК 512.57

Научная статья

Полный текст на русском языке

Получена 21 августа 2020 г.

После доработки 7 сентября 2020 г.

Принята к публикации 9 сентября 2020 г.

Рафаэль Робинсон показал, что все примитивно рекурсивные функции, зависящие от одного аргумента, и только они могут быть получены из двух функций $s(x) = x + 1$ и $q(x) = x - [\sqrt{x}]^2$ с помощью операций сложения $+$, суперпозиции $*$ и итерации i . Джулия Робинсон доказала, что из этих же двух функций с помощью операций сложения $+$, суперпозиции $*$ и операции $^{-1}$ обращения функций можно получить все общерекурсивные (при определённом условии на операцию обращения) и все частично рекурсивные функции. На основании этих результатов А. И. Мальцев ввёл в рассмотрение алгебру Рафаэля Робинсона всех одноместных примитивно рекурсивных функций и две алгебры Джулии Робинсон: частичную алгебру всех одноместных общерекурсивных функций и алгебру всех одноместных частично рекурсивных функций, и предложил исследовать свойства этих алгебр, в том числе, выяснить, существуют ли в этих алгебрах конечные базисы тождеств. В этой статье мы показываем, что конечного базиса тождеств ни в одной из указанных алгебр не существует.

Ключевые слова: алгебры; рекурсивные функции; тождества; базис; суперпозиция; итерация; обращение функции.

ИНФОРМАЦИЯ ОБ АВТОРАХ

Валерий Анатольевич Соколов
автор для корреспонденции

orcid.org/0000-0003-1427-4937. E-mail: valery-sokolov@yandex.ru
доктор физ.-мат. наук, профессор, кафедра теоретической информатики,
Центр интегрируемых систем.

Финансирование: Инициативная НИР ВИП-004 (номер госрегистрации AAAA-A16-116070610022-6).

Для цитирования: V. A. Sokolov, "On the Existence Problem of Finite Bases of Identities in the Algebras of Recursive Functions", *Modeling and analysis of information systems*, vol. 27, no. 3, pp. 304-315, 2020.

Введение

В данной работе мы будем рассматривать классы примитивно рекурсивных, общерекурсивных и частично рекурсивных функций, которые являются классами вычислимых функций, имеющими фундаментальное значение в математике [1, 2].

В 1947 году Рафаэль Робинсон в работе [3] доказал теорему о том, что все примитивно рекурсивные функции одной переменной, образующие класс $\mathfrak{F}_{\text{пр}}^{(1)}$, и только они могут быть получены из двух функций

$$s(x) = x + 1, \quad q(x) = x - [\sqrt{x}]^2 \quad (1)$$

с помощью операций сложения $+$, суперпозиции $*$ и итерации i , которые определяются следующим образом:

$$(f + g)(x) = f(x) + g(x), \quad (2)$$

$$(f * g)(x) = f(g(x)), \quad (3)$$

$$f = i(g), \text{ где } f(0) = 0 \text{ и } f(n+1) = g(f(n)), \quad n \geq 0. \quad (4)$$

Очевидно, итерация i является частным случаем примитивной рекурсии [2]. Результатами этих операций также являются одноместные примитивно рекурсивные функции из $\mathfrak{F}_{\text{пр}}^{(1)}$, т.е. класс $\mathfrak{F}_{\text{пр}}^{(1)}$ замкнут относительно указанных операций.

В 1950 году Джулия Робинсон в работе [4] доказала аналогичную теорему о том, что все общерекурсивные функции одной переменной, образующие класс $\mathfrak{F}_{\text{ор}}^{(1)}$, можно получить из тех же двух функций (1) с помощью операций сложения $+$, суперпозиции $*$ и операции обращения $^{-1}$, которая определяется следующим образом [2]:

$$f^{-1}(x) = \mu_y(f(y) = x), \quad (5)$$

где выражение $\mu_y(f(y) = x)$ определено и обозначает наименьшее значение y такое, что $f(y) = x$, лишь в том случае, если это значение y существует и при этом все значения $f(0), \dots, f(y-1)$ определены и не равны x . Очевидно, если требуется оставаться в классе $\mathfrak{F}_{\text{ор}}^{(1)}$, то операцию обращения $^{-1}$ следует считать частичной и применимой не ко всем функциям из $\mathfrak{F}_{\text{ор}}^{(1)}$, а только лишь к тем, которые в результате дают снова всюду определённую, то есть общерекурсивную функцию. Это частный случай так называемой слабой минимизации [2, стр. 49].

Результатами так определённых операций являются одноместные общерекурсивные функции из $\mathfrak{F}_{\text{ор}}^{(1)}$, т.е. класс $\mathfrak{F}_{\text{ор}}^{(1)}$ замкнут относительно указанных операций.

Если же разрешить применять операцию $^{-1}$ и в том случае, когда в результате может получаться частичная функция, то мы получим класс $\mathfrak{F}_{\text{чр}}^{(1)}$ всех одноместных частично рекурсивных функций. Этот класс $\mathfrak{F}_{\text{чр}}^{(1)}$, как следует из результата Дж. Робинсон [4], также порождается парой функций (1) с помощью операций $+$, $*$ и $^{-1}$, где операция $^{-1}$ применима уже ко всем функциям из $\mathfrak{F}_{\text{чр}}^{(1)}$.

В соответствии с этими результатами А. И. Мальцевым в 1961 году в работе [1] были введены три алгебры:

- 1) алгебра одноместных примитивно рекурсивных функций (алгебра ПРФ Рафаэля Робинсона)

$$\mathfrak{A}_{\text{пр}} = \langle \mathfrak{F}_{\text{пр}}^{(1)}; +, *, i \rangle; \quad (6)$$

- 2) алгебра одноместных общерекурсивных функций (алгебра ОРФ Джулии Робинсон)

$$\mathfrak{A}_{\text{ор}} = \langle \mathfrak{F}_{\text{ор}}^{(1)}; +, *, ^{-1} \rangle; \quad (7)$$

3) алгебра одноместных частично рекурсивных функций (*алгебра ЧРФ Джулии Робинсон*)

$$\mathfrak{A}_{\text{чр}} = \langle \mathfrak{F}_{\text{чр}}^{(1)}; +, *, ^{-1} \rangle \quad (8)$$

и поставлена задача изучения свойств этих алгебр, в том числе проблема существования (или отсутствия) конечных базисов тождеств в многообразиях алгебр, порождённых указанными алгебрами. Заметим, что, как следует из результатов Р. Робинсона и Дж. Робинсон [3, 4], каждая из алгебр $\mathfrak{A}_{\text{пр}}$, $\mathfrak{A}_{\text{ор}}$, $\mathfrak{A}_{\text{чр}}$ имеет конечный базис, состоящий из двух функций $s(x)$ и $q(x)$.

Цель данной статьи – показать, что ни в одном из многообразий $\mathfrak{M}_{\text{пр}}$, $\mathfrak{M}_{\text{ор}}$, $\mathfrak{M}_{\text{чр}}$, порождённых алгебрами $\mathfrak{A}_{\text{пр}}$, $\mathfrak{A}_{\text{ор}}$, $\mathfrak{A}_{\text{чр}}$ соответственно, конечного базиса тождеств не существует.

1. Алгебра ПРФ Рафаэля Робинсона

Пусть $\mathfrak{M}_{\text{пр}}$ – многообразие сигнатуры $\Omega_{\text{пр}} = \langle +, *, i \rangle$, порождённое алгеброй $\mathfrak{A}_{\text{пр}}$, и пусть $\mathfrak{A}$ – произвольная алгебра (той же сигнатуры $\Omega_{\text{пр}}$) из этого многообразия. Заметим, что так как по предположению алгебра $\mathfrak{A}$ принадлежит многообразию $\mathfrak{M}_{\text{пр}}$, в $\mathfrak{A}$ должны быть истинными все тождества многообразия $\mathfrak{M}_{\text{пр}}$.

Термы в алгебре $\mathfrak{A}$ определим следующим образом:

- 1) переменные v и v_n суть *термы* для любого натурального числа n ;
- 2) пусть t_1 и t_2 – *термы*, тогда $(t_1 + t_2)$, $(t_1 * t_2)$, $i(t_1)$ – *тоже термы*.

Для уменьшения количества скобок в записи термов будем придерживаться следующей иерархии операций: высший приоритет у унарной операции итерации i , затем идёт операция суперпозиции $*$ и, наконец, операция $+$ с самым низким приоритетом. Учитывая, что операции сложения и суперпозиции ассоциативны, а операция сложения коммутативна, замечаем, что любой терм t может быть представлен в виде

$$t_1 + t_2 + \dots + t_n, \quad (9)$$

где $n \geq 1$ и каждый терм t_j – это либо суперпозиция двух каких-то термов, либо итерация некоторого терма, либо простая переменная, $j = 1, 2, \dots, n$.

Если терм t не является суммой других термов, то такой терм назовём *одночленом*. Одночлен вида $i(t)$ назовём *i -замкнутым термом* (для любого терма t). Очевидно, любой одночлен – это либо суперпозиция нескольких термов $t_1 * \dots * t_k$, $k \geq 2$, либо i -замкнутый терм $i(t)$, либо простая переменная v (возможно, с индексом).

Терм $t = t_1 + t_2 + \dots + t_n$, $n \geq 1$, где каждый терм t_j , $1 \leq j \leq n$, является одночленом, будем называть *многочленом*, а число n – его *аддитивным рангом* $ar(t)$.

Пример: $ar(v_1 + v_1 * (v_2 + v_3 + v_4) + i v_1) = 3$.

Заметим, что если для некоторого терма t $ar(t) = 1$, то терм t является одночленом.

Значение терма $t(v_1, \dots, v_n)$, $n \geq 1$, в некоторой алгебре $\mathfrak{A}$ из многообразия $\mathfrak{M}_{\text{пр}}$, получающееся при подстановке вместо переменных v_j элементов a_j из основного множества алгебры $\mathfrak{A}$, обозначим

$$t(a_1, \dots, a_n)_{\mathfrak{A}}.$$

Тождество в алгебре $\mathfrak{A}$ многообразия $\mathfrak{M}_{\text{пр}}$ – это запись

$$t_1(v_1, \dots, v_n) = t_2(v_1, \dots, v_n),$$

где t_1 и t_2 – термы в алгебре $\mathfrak{A}$, каждый из которых, возможно, зависит фиктивно от некоторых переменных из списка $v_1, \dots, v_n$, $n \geq 1$, при этом

$$t_1(a_1, \dots, a_n)_{\mathfrak{A}} = t_2(a_1, \dots, a_n)_{\mathfrak{A}}$$

при любых значениях a_j (из носителя алгебры $\mathfrak{A}$) переменных v_j , $1 \leq j \leq n$.

Примеры тождеств в алгебре $\mathfrak{A}$:

$$v_1 + v_2 = v_2 + v_1;$$

$$i(iv_1 + iv_2) = i(iv_3).$$

Аддитивный ранг тождества $t_1 = t_2$ в алгебре $\mathfrak{A}$ многообразия $\mathfrak{M}_{\text{пр}}$ – это наибольший из аддитивных рангов термов t_1 и t_2 .

Теорема 1. В многообразии алгебр $\mathfrak{M}_{\text{пр}}$, порождённом алгеброй примитивно рекурсивных функций Рафаэля Робинсона $\mathfrak{A}_{\text{пр}} = \langle \mathfrak{F}_{\text{пр}}^{(1)}; +, *, i \rangle$, конечного базиса тождеств не существует.

Доказательство. Допустим, что в многообразии алгебр $\mathfrak{M}_{\text{пр}}$ существует конечный базис тождеств $\Sigma_{\text{пр}}$, т.е. конечный набор тождеств в сигнатуре $\Omega_{\text{пр}}$, из которых выводимы все тождества, истинные в любой алгебре многообразия алгебр $\mathfrak{M}_{\text{пр}}$.

Аддитивным рангом $ar(\Sigma_{\text{пр}})$ конечной системы тождеств $\Sigma_{\text{пр}}$ в многообразии $\mathfrak{M}_{\text{пр}}$ назовём максимальный аддитивный ранг тождеств, входящих в $\Sigma_{\text{пр}}$.

Так как в алгебре $\mathfrak{A}_{\text{пр}}$ справедливы тождества, выражающие коммутативность и ассоциативность операции сложения, это позволяет опускать лишние скобки в записи термов и игнорировать порядок слагаемых в суммах одночленов в записи термов в любой алгебре многообразия $\mathfrak{M}_{\text{пр}}$.

Далее рассмотрим ещё одну алгебру

$$\mathfrak{B}_{\text{пр}} = \langle \mathfrak{F}_{\text{пр}}^{(1)}; +, *, i \rangle, \quad (10)$$

у которой носитель $\mathfrak{F}_{\text{пр}}^{(1)}$ – то же самое множество всех одноместных примитивно рекурсивных функций, как и у алгебры $\mathfrak{A}_{\text{пр}}$, такая же сигнатура $\Omega_{\text{пр}}$, а определение термов совпадает с определением термов в алгебре $\mathfrak{A}$.

Все соглашения относительно старшинства операций в алгебре $\mathfrak{A}$ остаются в силе и для алгебры $\mathfrak{B}_{\text{пр}}$, равно как и замечание о строении термов. Таким образом, можно утверждать, что любой терм в алгебре $\mathfrak{B}_{\text{пр}}$ является многочленом либо одночленом, при этом любой многочлен является суммой двух или более одночленов.

Для алгебры $\mathfrak{B}_{\text{пр}}$ остаются справедливыми определения аддитивного ранга терма и аддитивного ранга тождества, так как они не зависят от семантики операций.

Что касается определений операций $+$ и $*$ в алгебре $\mathfrak{B}_{\text{пр}}$, то они точно такие же, как и для алгебры $\mathfrak{A}_{\text{пр}}$, а семантику операции i для алгебры $\mathfrak{B}_{\text{пр}}$ определим специальным образом.

Операция i в алгебре $\mathfrak{B}_{\text{пр}}$ отличается от аналогичной операции в алгебре $\mathfrak{A}_{\text{пр}}$ только для значений термов с «достаточно большими» аддитивными рангами.

Значения термов в алгебре $\mathfrak{A}_{\text{пр}}$.

1. Если терм t – простая предметная переменная v или v_n , то значением терма t в алгебре $\mathfrak{A}_{\text{пр}}$ будет f для любой функции f из $\mathfrak{F}_{\text{пр}}^{(1)}$.

2. Пусть $t_1(v_1, \dots, v_n)$ и $t_2(u_1, \dots, u_m)$ – термы. Тогда значениями термов

$$t_1(v_1, \dots, v_n) + t_2(u_1, \dots, u_m)$$

и

$$t_1(v_1, \dots, v_n) * t_2(u_1, \dots, u_m)$$

в алгебре $\mathfrak{A}_{\text{пр}}$ при подстановке $v_j = f_j$, $u_k = g_k$, где f_j, g_k – функции из $\mathfrak{F}_{\text{пр}}^{(1)}$, $1 \leq j \leq n$, $1 \leq k \leq m$, будут функции

$$t_1(f_1, \dots, f_n) + t_2(g_1, \dots, g_m)$$

и

$$t_1(f_1, \dots, f_n) * t_2(g_1, \dots, g_m)$$

соответственно.

3. Определим значение термина $i(t)$ для любого термина t в алгебре $\mathfrak{A}_{\text{пр}}$. Как было показано ранее, любой терм t алгебры из многообразия $\mathfrak{M}_{\text{пр}}$ является многочленом $t_1 + t_2 + \dots + t_n$, $n \geq 1$.

Предположим, что терм t зависит от переменных $v_1, \dots, v_k$, $k \geq 1$, и пусть функции $f_1, \dots, f_k$ из $\mathfrak{F}_{\text{пр}}^{(1)}$ являются значениями этих переменных. Можно считать, что каждый терм t_j зависит от всех переменных $v_1, \dots, v_k$ (возможно, что от некоторых из них t_j зависит фиктивно). Тогда значением термина $t(v_1, \dots, v_k) = i(t_1(v_1, \dots, v_k) + \dots + t_n(v_1, \dots, v_k))$ будет

$$t(f_1, \dots, f_k)_{\mathfrak{A}_{\text{пр}}} = i(t_1(f_1, \dots, f_k) + \dots + t_n(f_1, \dots, f_k)).$$

Значения термов в алгебре $\mathfrak{B}_{\text{пр}}$.

Пункты 1' и 2' определения значений термов в алгебре $\mathfrak{B}_{\text{пр}}$ дословно совпадают с пунктами 1 и 2 определения значений термов в алгебре $\mathfrak{A}_{\text{пр}}$.

3'. Определим теперь значение термина $i(t)$ для любого термина t в алгебре $\mathfrak{B}_{\text{пр}}$. Предположим, что терм t зависит от переменных $v_1, \dots, v_k$, $k \geq 1$, и пусть функции $f_1, \dots, f_k$ из $\mathfrak{F}_{\text{пр}}^{(1)}$ являются любыми значениями этих переменных. Терм t , как и в алгебре $\mathfrak{A}_{\text{пр}}$, будет суммой n одночленов:

$$t(v_1, \dots, v_k) = t_1(v_1, \dots, v_k) + \dots + t_n(v_1, \dots, v_k), \quad n \geq 1.$$

Тогда значением термина $i(t(v_1, \dots, v_k))_{\mathfrak{B}_{\text{пр}}}$ в алгебре $\mathfrak{B}_{\text{пр}}$ будет

$$\begin{aligned} i(t(f_1, \dots, f_k))_{\mathfrak{A}_{\text{пр}}}, & \text{ если } n \leq ar(\Sigma_{\text{пр}}), \\ i(t(f_1, \dots, f_k) + 1)_{\mathfrak{A}_{\text{пр}}}, & \text{ если } n > ar(\Sigma_{\text{пр}}). \end{aligned} \quad (11)$$

Отметим следующие очевидные утверждения, вытекающие из введенных определений.

1. Тождество $t_1 = t_2$, аддитивный ранг которого не превышает $ar(\Sigma_{\text{пр}})$, истинно в алгебре $\mathfrak{A}_{\text{пр}}$ тогда и только тогда, когда оно истинно в алгебре $\mathfrak{B}_{\text{пр}}$.

2. Конечная система тождеств $\Sigma_{\text{пр}}$, истинная в алгебре $\mathfrak{A}_{\text{пр}}$, истинна и в алгебре $\mathfrak{B}_{\text{пр}}$.

Заметим, что любые следствия из тождеств системы $\Sigma_{\text{пр}}$ должны быть истинны в любой алгебре многообразия $\mathfrak{M}_{\text{пр}}$, включая алгебру $\mathfrak{B}_{\text{пр}}$.

Очевидно, все термы t_1 и t_2 , входящие в тождества $t_1 = t_2$ системы $\Sigma_{\text{пр}}$, имеют в алгебре $\mathfrak{B}_{\text{пр}}$ такие же значения, как и в алгебре $\mathfrak{A}_{\text{пр}}$, то есть, как было сказано, алгебра $\mathfrak{B}_{\text{пр}}$ принадлежит многообразию $\mathfrak{M}_{\text{пр}}$. Это означает, что все тождества, выводимые из тождеств системы $\Sigma_{\text{пр}}$, должны быть истинными как в алгебре $\mathfrak{A}_{\text{пр}}$, так и в алгебре $\mathfrak{B}_{\text{пр}}$.

Рассмотрим серию тождеств в алгебре $\mathfrak{A}_{\text{пр}}$ вида

$$i(iv_1 + \dots + iv_n) = i(iv_{n+1}), \quad n \geq 1. \quad (12)$$

Это так называемые «нулевые» тождества [5]. Нетрудно видеть, что при любых значениях переменных $v_1, \dots, v_{n+1}$ и для любого натурального числа $n \geq 1$ значения термов в левой и в правой частях выражения (12) равны функции-константе 0.

Тождества данной серии истинны в алгебре $\mathfrak{A}_{\text{пр}}$, а значит, должны быть истинны в любой алгебре многообразия $\mathfrak{M}_{\text{пр}}$.

Однако, равенство $i(iv_1 + \dots + iv_n) = i(iv_{n+1})$ при $n > ar(\Sigma_{\text{пр}})$, очевидно, является тождеством в алгебре $\mathfrak{A}_{\text{пр}}$, но в алгебре $\mathfrak{B}_{\text{пр}}$ в этом случае для любых $f_1, \dots, f_{n+1}$ из $\mathfrak{F}_{\text{пр}}^{(1)}$ мы имеем в левой части выражения (12) значение термина

$$i(if_1 + \dots + if_n + 1) \neq 0,$$

а в правой части выражения (12) мы имеем 0, так как для любой функции f из $\mathfrak{F}_{\text{пр}}^{(1)}$ значение $i(if)_{\mathfrak{B}_{\text{пр}}} = 0$.

Таким образом, видим, что выражение

$$i(iv_1 + \dots + iv_n) = i(iv_{n+1})$$

при $n > ar(\Sigma_{\text{пр}})$ не является тождеством в алгебре $\mathfrak{B}_{\text{пр}}$.

Полученное противоречие означает, что предположение о существовании конечного базиса тождеств $\Sigma_{\text{пр}}$ в многообразии алгебр $\mathfrak{M}_{\text{пр}}$ (а значит, и в алгебре $\mathfrak{A}_{\text{пр}}$) является неверным, что и требовалось доказать. $\square$

Следствие 1. В алгебре Рафаэля Робинсона одноместных примитивно рекурсивных функций $\mathfrak{A}_{\text{пр}} = \langle \mathfrak{F}_{\text{пр}}^{(1)}; +, *, i \rangle$ конечного базиса тождеств не существует.

Следствие 2. В алгебре $\mathfrak{A}_{\text{пр}}^+ = \langle \mathfrak{F}_{\text{пр}}^{(1)}; +, i \rangle$ конечного базиса тождеств не существует.

Обозначим через $\mathfrak{F}^{(1)}$ множество всех одноместных теоретико-числовых функций, определённых на множестве натуральных чисел и со значениями в том же множестве.

Следствие 3. В алгебрах одноместных теоретико-числовых функций $\langle \mathfrak{F}^{(1)}; +, *, i \rangle$ и $\langle \mathfrak{F}^{(1)}; +, i \rangle$ конечного базиса тождеств не существует.

2. Алгебра ЧРФ Джулии Робинсон

Теперь перейдём к алгебре всех одноместных частично рекурсивных функций $\mathfrak{A}_{\text{чр}} = \langle \mathfrak{F}_{\text{чр}}^{(1)}; +, *, ^{-1} \rangle$. Операции сложения $+$ и суперпозиции $*$ частично рекурсивных функций в алгебре $\mathfrak{A}_{\text{чр}}$ определяются аналогично соответствующим операциям (2), (3) алгебры $\mathfrak{A}_{\text{пр}}$, при этом операция обращения $^{-1}$ в алгебре $\mathfrak{A}_{\text{чр}}$, в отличие от алгебры $\mathfrak{A}_{\text{ор}}$, будет всюду определённой, так как результатом применения этой операции к любой частично рекурсивной функции будет некоторая частично рекурсивная функция из алгебры $\mathfrak{A}_{\text{чр}}$.

Схема доказательства для этого случая похожа на рассуждения в Теореме 1, но имеет некоторые особенности.

Теорема 2. В многообразии алгебр $\mathfrak{M}_{\text{чр}}$, порождённом алгеброй частично рекурсивных функций Джулии Робинсон $\mathfrak{A}_{\text{чр}} = \langle \mathfrak{F}_{\text{чр}}^{(1)}; +, *, ^{-1} \rangle$, конечного базиса тождеств не существует.

Доказательство. Пусть $\mathfrak{M}_{\text{чр}}$ – многообразие алгебр сигнатуры $\Omega_{\text{чр}} = \langle +, *, ^{-1} \rangle$, порождённое алгеброй $\mathfrak{A}_{\text{чр}}$, и пусть $\mathfrak{G}$ – произвольная алгебра из этого многообразия.

Термы в алгебре $\mathfrak{G}$ определяются обычным образом:

- 1) переменные v и v_n суть *термы* для любого натурального числа n ;
- 2) пусть t_1 и t_2 – *термы*, тогда $(t_1 + t_2)$, $(t_1 * t_2)$, $(t_1)^{-1}$ – тоже *термы*.

Как и в Теореме 1, в записи термов будем придерживаться следующей иерархии операций: высший приоритет у одноместной операции обращения функции $^{-1}$, затем идёт операция суперпозиции $*$ и, наконец, операция $+$ с самым низким приоритетом. Очевидно, любой терм t в алгебре $\mathfrak{A}_{\text{чр}}$ может быть представлен в виде (9):

$$t_1 + t_2 + \dots + t_n,$$

где $n \geq 1$ и каждый терм t_j – это либо суперпозиция двух каких-то термов, либо обращение некоторого терма, либо простая переменная, $j = 1, 2, \dots, n$.

Определим *ранг терма* t по суперпозиции $sr(t)$:

- 1) $sr(v) = 0$, где v – простая переменная (возможно, с индексом);
- 2) пусть t_1 и t_2 – термы, тогда

$$sr(t_1 + t_2) = sr(t_1) + sr(t_2),$$

$$sr(t_1 * t_2) = sr(t_1) + sr(t_2) + 1,$$

$$sr((t_1)^{-1}) = sr(t_1).$$

Заметим, что если для некоторого терма t $sr(t) = n$, то в записи терма t присутствуют ровно n символов операции суперпозиции $*$, $n \geq 0$.

Пример: $sr(v_1 + (v_1 * v_2) * v_1 + (v_2 * v_1)^{-1}) = 3$.

Значение терма $t(v_1, \dots, v_n)$, $n \geq 1$, в некоторой алгебре $\mathfrak{G}$ из многообразия $\mathcal{M}_{\text{чр}}$, получающееся при подстановке вместо переменных v_j элементов a_j из основного множества алгебры $\mathfrak{G}$, как и прежде, будем обозначать

$$t(a_1, \dots, a_n)_{\mathfrak{G}}.$$

Тождества в алгебре $\mathfrak{G}$ многообразия $\mathcal{M}_{\text{чр}}$ определяются обычным образом.

Примеры тождеств в алгебре $\mathfrak{G}$

$$(v_1 + v_2) + v_3 = v_1 + (v_2 + v_3); \quad (13)$$

$$(v_1 * v_2) * v_3 = v_1 * (v_2 * v_3); \quad (14)$$

$$v_1 + v_2 = v_2 + v_1; \quad (15)$$

$$v^{-1} * v * v^{-1} = v^{-1}. \quad (16)$$

Ранг по суперпозиции тождества $t_1 = t_2$ в многообразии $\mathcal{M}_{\text{чр}}$ – это наибольший из рангов по суперпозиции $sr(t_1)$ и $sr(t_2)$.

Допустим, что в многообразии алгебр $\mathcal{M}_{\text{чр}}$ существует *конечный базис тождеств* $\Sigma_{\text{чр}}$, т.е. конечный набор тождеств в сигнатуре $\Omega_{\text{чр}}$, из которых выводимы все тождества, истинные в любой алгебре многообразия алгебр $\mathcal{M}_{\text{чр}}$.

Рангом по суперпозиции $sr(\Sigma_{\text{чр}})$ конечной системы тождеств $\Sigma_{\text{чр}}$ в многообразии алгебр $\mathcal{M}_{\text{чр}}$ назовём максимальный ранг по суперпозиции тождеств, входящих в $\Sigma_{\text{чр}}$.

Так как в алгебре $\mathcal{A}_{\text{чр}}$ справедливы тождества, выражающие коммутативность операции сложения (15) и ассоциативность операций сложения (13) и суперпозиции (14), то это, как и ранее, позволяет опускать лишние скобки в записи термов в любой алгебре многообразия $\mathcal{M}_{\text{чр}}$.

Рассмотрим ещё одну алгебру

$$\mathfrak{B}_{\text{чр}} = \langle \tilde{\mathfrak{F}}_{\text{чр}}^{(1)}; +, *, ^{-1} \rangle, \quad (17)$$

у которой носитель $\tilde{\mathfrak{F}}_{\text{чр}}^{(1)}$ – то же самое множество всех одноместных частично рекурсивных функций, как и у алгебры $\mathcal{A}_{\text{чр}}$, такая же сигнатура $\Omega_{\text{чр}}$, а определение термов совпадает с определением термов в алгебре $\mathfrak{G}$.

Все соглашения относительно старшинства операций в алгебре $\mathfrak{G}$ остаются в силе и для алгебры $\mathfrak{B}_{\text{чр}}$, а также остаются справедливыми определения *ранга по суперпозиции терма* и *ранга по суперпозиции тождества*, так как они не зависят от семантики операций.

Что касается определений операций $+$ и $*$ в алгебре $\mathfrak{B}_{\text{чр}}$, то они точно такие же, как и для алгебры $\mathcal{A}_{\text{чр}}$, а семантику операции обращения $^{-1}$ для алгебры $\mathfrak{B}_{\text{чр}}$ определим специальным образом.

Значения термов в алгебре $\mathcal{A}_{\text{чр}}$ определяются стандартным образом, аналогично определениям для алгебры $\mathcal{A}_{\text{пр}}$ (с заменой носителя алгебры, а также с заменой символа операции i на символ операции $^{-1}$).

Значения термов в алгебре $\mathfrak{B}_{\text{чр}}$.

Определения значений термов для простой переменной v (возможно, с индексом), а также для термов вида $t_1 + t_2$ и $t_1 * t_2$, в алгебре $\mathfrak{B}_{\text{чр}}$ такие же, как и определения значений термов для аналогичных операций в алгебре $\mathfrak{A}_{\text{чр}}$.

Определим теперь значение терма t^{-1} для любого терма t в алгебре $\mathfrak{B}_{\text{чр}}$. Предположим, что терм t зависит от переменных $v_1, \dots, v_k$, $k \geq 1$, и пусть функции $f_1, \dots, f_k$ из $\mathfrak{F}_{\text{чр}}^{(1)}$ являются любыми значениями этих переменных. Тогда значением терма $(t^{-1}(v_1, \dots, v_k))_{\mathfrak{B}_{\text{чр}}}$ в алгебре $\mathfrak{B}_{\text{чр}}$ будет

$$\begin{aligned} & ((t(f_1, \dots, f_k))^{-1})_{\mathfrak{A}_{\text{чр}}}, \text{ если } sr(t) \leq sr(\Sigma_{\text{чр}}), \\ & ((t(f_1, \dots, f_k) + 1)^{-1})_{\mathfrak{A}_{\text{чр}}}, \text{ если } sr(t) > sr(\Sigma_{\text{чр}}). \end{aligned} \quad (18)$$

Отметим следующие очевидные утверждения, вытекающие из введенных определений.

1. Тождество $t_1 = t_2$, ранг по суперпозиции которого не превышает $sr(\Sigma_{\text{чр}})$, истинно в алгебре $\mathfrak{A}_{\text{чр}}$ тогда и только тогда, когда оно истинно в алгебре $\mathfrak{B}_{\text{чр}}$.

2. Конечная система тождеств $\Sigma_{\text{чр}}$, истинная в алгебре $\mathfrak{A}_{\text{чр}}$, истинна и в алгебре $\mathfrak{B}_{\text{чр}}$.

Заметим, что любые следствия из тождеств системы $\Sigma_{\text{чр}}$ должны быть истинны в любой алгебре многообразия $\mathfrak{M}_{\text{чр}}$, включая алгебру $\mathfrak{B}_{\text{чр}}$.

Очевидно, все термы t_1 и t_2 , входящие в тождества $t_1 = t_2$ системы $\Sigma_{\text{чр}}$, имеют в алгебре $\mathfrak{B}_{\text{чр}}$ такие же значения, как и в алгебре $\mathfrak{A}_{\text{чр}}$, то есть алгебра $\mathfrak{B}_{\text{чр}}$ принадлежит многообразию $\mathfrak{M}_{\text{чр}}$. Это означает, что все тождества, выводимые из тождеств системы $\Sigma_{\text{чр}}$, должны быть истинными как в алгебре $\mathfrak{A}_{\text{чр}}$, так и в алгебре $\mathfrak{B}_{\text{чр}}$.

Построим бесконечную серию тождеств в алгебре $\mathfrak{A}_{\text{чр}}$. Пусть термы t_n определены по индукции следующим образом:

$$\begin{aligned} t_0(v) &= v^{-1}; \\ t_{n+1}(v) &= t_n(v) * v * v^{-1}, \quad n \geq 1. \end{aligned} \quad (19)$$

Тогда для каждого $n \geq 0$ в алгебре $\mathfrak{A}_{\text{чр}}$ имеем тождество

$$(t_n(v) * v * v^{-1})^{-1} = (v^{-1})^{-1}. \quad (20)$$

Нетрудно видеть, в частности, что при $n = 0$ – это тождество

$$(v^{-1} * v * v^{-1})^{-1} = (v^{-1})^{-1}, \quad (21)$$

получающееся из тождества (16), которое приведено в [2] на стр. 50, а

$$sr(t_n(v)) = 2n, \quad n \geq 0.$$

Тождества серии (20) истинны в алгебре $\mathfrak{A}_{\text{чр}}$, а значит, должны быть истинны в любой алгебре многообразия алгебр $\mathfrak{M}_{\text{чр}}$.

Однако, при $sr(t_n(v)) > sr(\Sigma_{\text{чр}})$ равенство $(t_n(v) * v * v^{-1})^{-1} = (v^{-1})^{-1}$ является тождеством в алгебре $\mathfrak{A}_{\text{чр}}$ для любого $n \geq 0$, а в алгебре $\mathfrak{B}_{\text{чр}}$ в этом случае для произвольной функции f из $\mathfrak{F}_{\text{чр}}^{(1)}$ мы имеем в левой части выражения (20) значение терма $(t_n(f) * f * f^{-1} + 1)^{-1}$, а в правой части выражения (20) мы имеем $(f^{-1})^{-1}$, т.е. равенство (20) в этом случае нарушается в алгебре $\mathfrak{B}_{\text{чр}}$ (например, для $f(x) = x$).

Таким образом, видим, что выражение

$$(t_n(v) * v * v^{-1})^{-1} = (v^{-1})^{-1}$$

при $2(n+1) > sr(\Sigma_{\text{чр}})$ не является тождеством в алгебре $\mathfrak{B}_{\text{чр}}$.

Полученное противоречие означает, что предположение о существовании конечного базиса тождеств $\Sigma_{\text{чр}}$ в многообразии алгебр $\mathfrak{M}_{\text{чр}}$ (а значит, и алгебре $\mathfrak{A}_{\text{чр}}$) является неверным, что и требовалось доказать. $\square$

Следствие 4. В алгебре Джулии Робинсон одноместных частично рекурсивных функций $\mathfrak{A}_{\text{чр}} = \langle \mathfrak{F}_{\text{чр}}^{(1)}; +, *, ^{-1} \rangle$ конечного базиса тождеств не существует.

Следствие 5. В алгебре $\mathfrak{A}_{\text{чр}}^* = \langle \mathfrak{F}_{\text{чр}}^{(1)}; *, ^{-1} \rangle$ конечного базиса тождеств не существует.

Обозначим через $\mathfrak{F}_{\text{ч}}^{(1)}$ множество всех одноместных частичных теоретико-числовых функций, определённых на подмножествах множества натуральных чисел и со значениями в множестве натуральных чисел.

Следствие 6. В алгебрах одноместных частичных теоретико-числовых функций $\langle \mathfrak{F}_{\text{ч}}^{(1)}; +, *, ^{-1} \rangle$ и $\langle \mathfrak{F}_{\text{ч}}^{(1)}; *, ^{-1} \rangle$ конечного базиса тождеств не существует.

3. Алгебра ОРФ Джулии Робинсон

В заключение рассмотрим алгебру всех одноместных общерекурсивных функций Джулии Робинсон $\mathfrak{A}_{\text{ор}} = \langle \mathfrak{F}_{\text{ор}}^{(1)}; +, *, ^{-1} \rangle$. Особенность этого случая в том, что алгебра $\mathfrak{A}_{\text{ор}}$ является частичной [1, 6], так как операция обращения функции $^{-1}$ применима лишь к таким общерекурсивным функциям, которые в результате применения этой операции дают всюду определённые функции, т.е. операция $^{-1}$ является частичной. Для частичной алгебры $\mathfrak{A}_{\text{ор}}$ значение терма $t(v_1, \dots, v_n)$, $n \geq 1$, может быть неопределённым для некоторого набора значений переменных $a_1, \dots, a_n$ из основного множества алгебры $\mathfrak{A}_{\text{ор}}$. Соответственно, требуется уточнить понятие тождества, которое будет использоваться в этом случае.

Следуя С. Клини [7], тождеством в частичной алгебре $\mathfrak{A}_{\text{ор}}$ будем называть запись

$$t_1(v_1, \dots, v_n) \approx t_2(v_1, \dots, v_n),$$

где t_1 и t_2 – термы в алгебре $\mathfrak{A}_{\text{ор}}$, каждый из которых, возможно, зависит фиктивно от некоторых переменных из списка $v_1, \dots, v_n$, $n \geq 1$, такие, что при любых значениях a_j (из носителя алгебры $\mathfrak{A}_{\text{ор}}$) переменных v_j , $1 \leq j \leq n$, либо значения $t_1(a_1, \dots, a_n)_{\mathfrak{G}}$ и $t_2(a_1, \dots, a_n)_{\mathfrak{G}}$ определены и равны, либо оба значения термов $t_1(a_1, \dots, a_n)_{\mathfrak{G}}$ и $t_2(a_1, \dots, a_n)_{\mathfrak{G}}$ не определены.

Примеры тождеств в частичной алгебре $\mathfrak{A}_{\text{ор}}$:

$$(v_1 * v_2) * v_3 \approx v_1 * (v_2 * v_3), \quad (22)$$

$$v^{-1} * v * v^{-1} \approx v^{-1}. \quad (23)$$

Нетрудно убедиться в истинности в частичной алгебре $\mathfrak{A}_{\text{ор}}$ этих тождеств в семантике Клини.

Пусть $\mathfrak{G}$ – произвольная частичная алгебра из многообразия $\mathfrak{M}_{\text{ор}}$, порождённого частичной алгеброй $\mathfrak{A}_{\text{ор}} = \langle \mathfrak{F}_{\text{ор}}^{(1)}; +, *, ^{-1} \rangle$.

Термы в алгебре $\mathfrak{G}$ определяются так же, как и в алгебре $\mathfrak{A}_{\text{чр}}$, при этом любой терм t в алгебре $\mathfrak{G}$ может быть представлен, как и в алгебре $\mathfrak{A}_{\text{чр}}$, в виде

$$t_1 + t_2 + \dots + t_n,$$

где $n \geq 1$ и каждый терм t_j – это либо суперпозиция двух каких-то термов, либо обращение $^{-1}$ некоторого терма, либо простая переменная, $j = 1, 2, \dots, n$.

Определение *ранга терма* t по суперпозиции $sr(t)$ в алгебре $\mathfrak{G}$ совпадает с соответствующим определением в алгебре $\mathfrak{A}_{\text{чр}}$, так же как и определение *ранга по суперпозиции тождества* $t_1 \approx t_2$ в многообразии $\mathfrak{M}_{\text{ор}}$.

Как и в случае многообразия $\mathfrak{M}_{\text{чр}}$, *рангом по суперпозиции* $sr(\Sigma_{\text{ор}})$ *конечной системы тождеств* $\Sigma_{\text{ор}}$ в многообразии алгебр $\mathfrak{M}_{\text{ор}}$ назовём максимальный ранг по суперпозиции тождеств, входящих в $\Sigma_{\text{ор}}$.

Будем опускать лишние скобки в записи термов в любой алгебре многообразия $\mathfrak{M}_{\text{ор}}$, пользуясь ассоциативностью операций $+$ и $*$ и коммутативностью операции $+$.

Дадим определение *значения терма* t^{-1} для любого терма t в частичной алгебре $\mathfrak{A}_{\text{ор}}$.

Предположим, что терм t зависит от переменных $v_1, \dots, v_k$, $k \geq 1$, и пусть функции $f_1, \dots, f_k$ из $\mathfrak{F}_{\text{ор}}^{(1)}$ являются произвольными значениями этих переменных.

Тогда *значением терма* $t^{-1}(v_1, \dots, v_k)$ в частичной алгебре $\mathfrak{A}_{\text{ор}}$ будет функция $g = (t(f_1, \dots, f_k))^{-1}$, если она всюду определена на множестве натуральных чисел (т.е. является общерекурсивной функцией), а если это не так, то значение терма t^{-1} в алгебре $\mathfrak{A}_{\text{ор}}$ считается неопределённым на данном наборе $f_1, \dots, f_k$ значений переменных $v_1, \dots, v_k$.

Теорема 3. В многообразии частичных алгебр $\mathfrak{M}_{\text{ор}}$, порождённом частичной алгеброй одноместных общерекурсивных функций Джулии Робинсон $\mathfrak{A}_{\text{ор}} = \langle \mathfrak{F}_{\text{ор}}^{(1)}; +, *, {}^{-1} \rangle$, *конечного базиса тождеств* (в определённом выше смысле) не существует.

Доказательство. Предположим, что в многообразии частичных алгебр $\mathfrak{M}_{\text{ор}}$ существует *конечный базис тождеств* $\Sigma_{\text{ор}}$, т.е. конечный набор тождеств в сигнатуре $\Omega_{\text{ор}}$, из которых выводимы все тождества, истинные в любой алгебре многообразия $\mathfrak{M}_{\text{ор}}$.

Рассмотрим частичную алгебру

$$\mathfrak{B}_{\text{ор}} = \langle \mathfrak{F}_{\text{ор}}^{(1)}; +, *, {}^{-1} \rangle, \quad (24)$$

у которой носитель $\mathfrak{F}_{\text{ор}}^{(1)}$ – множество всех одноместных общерекурсивных функций, как и у алгебры $\mathfrak{A}_{\text{ор}}$, такая же сигнатура $\Omega_{\text{ор}}$, а определение термов совпадает с определением термов в алгебре $\mathfrak{G}$.

Все соглашения относительно старшинства операций в алгебре $\mathfrak{G}$ остаются в силе и для алгебры $\mathfrak{B}_{\text{ор}}$, а также остаются справедливыми определения *ранга терма по суперпозиции* и *ранга тождества по суперпозиции*. Что касается определений операций $+$ и $*$ в частичных алгебрах $\mathfrak{A}_{\text{ор}}$ и $\mathfrak{B}_{\text{ор}}$, то они такие же, как и для алгебры $\mathfrak{A}_{\text{чр}}$, а семантику операции обращения ${}^{-1}$ для алгебры $\mathfrak{B}_{\text{ор}}$ определим специальным образом.

Пусть терм t зависит от переменных $v_1, \dots, v_k$, $k \geq 1$, и пусть функции $f_1, \dots, f_k$ из $\mathfrak{F}_{\text{ор}}^{(1)}$ являются значениями этих переменных.

Тогда *значением терма* $(t^{-1}(v_1, \dots, v_k))_{\mathfrak{B}_{\text{ор}}}$ для данных значений $f_1, \dots, f_k$ переменных $v_1, \dots, v_k$ в алгебре $\mathfrak{B}_{\text{ор}}$ будет:

$$\begin{aligned} & ((t(f_1, \dots, f_k))^{-1})_{\mathfrak{A}_{\text{ор}}}, \text{ если } sr(t) \leq sr(\Sigma_{\text{ор}}), \\ & ((t(f_1, \dots, f_k) + 1)^{-1})_{\mathfrak{A}_{\text{ор}}}, \text{ если } sr(t) > sr(\Sigma_{\text{ор}}). \end{aligned} \quad (25)$$

Отметим следующие очевидные утверждения, вытекающие из введённых определений.

1. Тождество $t_1 \approx t_2$, ранг по суперпозиции которого не превышает $sr(\Sigma_{\text{ор}})$, истинно в частичной алгебре $\mathfrak{A}_{\text{ор}}$ тогда и только тогда, когда оно истинно в частичной алгебре $\mathfrak{B}_{\text{ор}}$.

2. Конечная система тождеств $\Sigma_{\text{ор}}$, истинная в частичной алгебре $\mathfrak{A}_{\text{ор}}$, истинна и в частичной алгебре $\mathfrak{B}_{\text{ор}}$.

Очевидно, все термы t_1 и t_2 , входящие в тождества $t_1 \approx t_2$ системы $\Sigma_{\text{ор}}$, имеют в алгебре $\mathfrak{B}_{\text{ор}}$ такие же значения, как и в алгебре $\mathfrak{A}_{\text{ор}}$, то есть алгебра $\mathfrak{B}_{\text{ор}}$, очевидно, принадлежит многообразию $\mathfrak{M}_{\text{ор}}$.

Это означает, что все тождества, выводимые из тождеств системы $\Sigma_{\text{ор}}$, должны быть истинными как в алгебре $\mathfrak{A}_{\text{ор}}$, так и в алгебре $\mathfrak{B}_{\text{ор}}$.

Рассмотрим серию равенств $t_n(v) * v * v^{-1} \approx v^{-1}$, $n \geq 0$, в произвольной частичной алгебре $\mathfrak{G}$ многообразия $\mathfrak{M}_{\text{ор}}$, где термы $t_n(v)$ определены по формулам (19), и $sr(t_n(v)) = 2n$, $n \geq 0$.

Нетрудно видеть, что эти равенства, а также их следствия

$$(t_n(v) * v * v^{-1})^{-1} \approx (v^{-1})^{-1}, n \geq 0, \quad (26)$$

являются тождествами в семантике Клини в частичной алгебре $\mathfrak{A}_{\text{ор}}$ и, следовательно, должны быть истинны в любой алгебре многообразия $\mathfrak{M}_{\text{ор}}$.

Но в алгебре $\mathfrak{B}_{\text{ор}}$, в соответствии с определением (25), при $sr(t_n(v)) > sr(\Sigma_{\text{ор}})$ равенство (26) нарушается, например, для функции $f(x) = x$ из $\mathfrak{F}_{\text{ор}}^{(1)}$. Таким образом, видим, что выражения $(t_n(v) * v * v^{-1})^{-1} \approx (v^{-1})^{-1}$ при $2(n+1) > sr(\Sigma_{\text{ор}})$ не являются тождествами в частичной алгебре $\mathfrak{B}_{\text{ор}}$.

Полученное противоречие означает, что предположение о существовании конечного базиса тождеств $\Sigma_{\text{ор}}$ в многообразии частичных алгебр $\mathfrak{M}_{\text{ор}}$ (а значит, и в частичной алгебре $\mathfrak{A}_{\text{ор}}$) является неверным, что и требовалось доказать. $\square$

Следствие 7. В частичной алгебре Джулии Робинсон одноместных общерекурсивных функций $\mathfrak{A}_{\text{ор}} = \langle \mathfrak{F}_{\text{ор}}^{(1)}; +, *, {}^{-1} \rangle$ конечного базиса тождеств не существует.

Следствие 8. В частичной алгебре $\mathfrak{A}_{\text{ор}}^* = \langle \mathfrak{F}_{\text{ор}}^{(1)}; *, {}^{-1} \rangle$ конечного базиса тождеств не существует.

Следствие 9. В алгебрах одноместных теоретико-числовых функций $\langle \mathfrak{F}_{\text{ор}}^{(1)}; +, *, {}^{-1} \rangle$ и $\langle \mathfrak{F}_{\text{ор}}^{(1)}; *, {}^{-1} \rangle$ конечного базиса тождеств не существует.

Заключение

В работе решена проблема А. И. Мальцева существования конечного базиса тождеств в алгебрах одноместных рекурсивных функций. Показано, что в алгебрах примитивно рекурсивных и частично рекурсивных функций конечного базиса тождеств не существует. Так как алгебра одноместных общерекурсивных функций является частичной, для неё аналогичная проблема была решена в семантике Клини для тождеств.

References

- [1] A. I. Mal'tsev, "Constructive algebras I", *Russian Mathematical Surveys*, vol. 16, no. 3, pp. 77–129, 1961.
- [2] A. I. Mal'tsev, *Algoritmy i rekursivnye funktsii*. Moscow: Nauka, 1965, In Russian.
- [3] J. Robinson, "Primitive recursive functions", *Proceedings of the American Mathematical Society*, vol. 1, no. 6, pp. 703–718, 1950.
- [4] R. M. Robinson, "Primitive recursive functions", *Bulletin of the American Mathematical Society*, vol. 53, no. 10, pp. 925–942, 1947.
- [5] V. A. Sokolov, "Ob odnom klasse tozhdestv v algebre Robinsona", in *14-ya Vsesoyuznaya algebraicheskaya konferentsiya: tezisy dokladov*, In Russian, vol. 2, Novosibirsk, 1977, pp. 123–124.
- [6] P. M. Cohn, *Universal Algebra*. New York, Evanston, and London: Harper & Row, 1965.
- [7] A. Robinson, "Equational logic for partial functions under Kleene equality: a complete and an incomplete set of rules", *The Journal of Symbolic Logic*, vol. 54, no. 2, pp. 354–362, 1989.

Application of Convolutional Neural Networks for Recognizing Long Structural Elements of Rails in Eddy-Current Defectograms

E. V. Kuzmin¹, O. E. Gorbunov², P. O. Plotnikov², V. A. Tyukin², V. A. Bashkin¹

DOI: [10.18255/1818-1015-2020-3-316-329](https://doi.org/10.18255/1818-1015-2020-3-316-329)

¹P. G. Demidov Yaroslavl State University, 14 Sovetskaya str., Yaroslavl 150003, Russia.

²Center of Innovative Programming, NDDLab, 144 Soyuznaya str., Yaroslavl, 150008, Russia.

MSC2020: 68T07

Research article

Full text in Russian

Received July 20, 2020

After revision August 3, 2020

Accepted September 9, 2020

To ensure traffic safety of railway transport, non-destructive test of rails is regularly carried out by using various approaches and methods, including eddy-current flaw detection methods. An automatic analysis of large data sets (defectograms) that come from the corresponding equipment is an actual problem. The analysis means a process of determining the presence of defective sections along with identifying structural elements of railway tracks in defectograms. This article is devoted to the problem of recognizing images of long structural elements of rails in eddy-current defectograms. Two classes of rail track structural elements are considered: 1) rolling stock axle counters, 2) rail crossings. Long marks that cannot be assigned to these two classes are conditionally considered as defects and are placed in a separate third class. For image recognition of structural elements in defectograms a convolutional neural network is applied. The neural network is implemented by using the open library TensorFlow. To this purpose each selected (picked out) area of a defectogram is converted into a graphic image in a grayscale with size of 30 x 140 points.

Keywords: nondestructive testing; eddy-current testing; rail flaw detection; automated analysis of defectograms; neural networks.

INFORMATION ABOUT THE AUTHORS

Egor V. Kuzmin correspondence author	orcid.org/0000-0003-0500-306X . E-mail: kuzmin@uniyar.ac.ru , kuzminev@nddlab.com Professor, Doctor of Science.
Oleg E. Gorbunov	orcid.org/0000-0001-6274-9971 . E-mail: gorbunovoe@nddlab.com General Director, PhD.
Petr O. Plotnikov	orcid.org/0000-0001-5687-7969 . E-mail: plotnikovpo@nddlab.com Production Engineer.
Vadim A. Tyukin	orcid.org/0000-0001-9149-7435 . E-mail: tyukinva@nddlab.com Head of Software Development.
Vladimir A. Bashkin	orcid.org/0000-0002-2534-1026 . E-mail: bashkinva@nddlab.com Professor, Doctor of Science.

For citation: E. V. Kuzmin, O. E. Gorbunov, P. O. Plotnikov, V. A. Tyukin, and V. A. Bashkin, "Application of Convolutional Neural Networks for Recognizing Long Structural Elements of Rails in Eddy-Current Defectograms", *Modeling and analysis of information systems*, vol. 27, no. 3, pp. 316-329, 2020.

Применение свёрточных нейронных сетей для распознавания длинных конструктивных элементов рельсов на вихретоковых дефектограммах

Е. В. Кузьмин¹, О. Е. Горбунов², П. О. Плотников², В. А. Тюкин², В. А. Башкин¹

DOI: [10.18255/1818-1015-2020-3-316-329](https://doi.org/10.18255/1818-1015-2020-3-316-329)

¹Ярославский государственный университет им. П. Г. Демидова, ул. Советская, 14, г. Ярославль, 150003 Россия.

²ООО «Центр инновационного программирования», NDDLab, ул. Союзная, 144, г. Ярославль, 150008 Россия.

УДК 004.032.26

Научная статья

Полный текст на русском языке

Получена 20 июля 2020 г.

После доработки 3 августа 2020 г.

Принята к публикации 9 сентября 2020 г.

Для обеспечения безопасности движения на железнодорожном транспорте регулярно проводится неразрушающий контроль рельсов с применением различных подходов и методов, включая методы вихретоковой дефектоскопии. Актуальной задачей является автоматический анализ больших массивов данных (дефектограмм), которые поступают от соответствующего оборудования. Под анализом понимается процесс определения по дефектограммам наличия дефектных участков наряду с выявлением конструктивных элементов рельсового пути. Данная статья посвящена задаче распознавания образов длинных конструктивных элементов железнодорожных рельсов по дефектограммам многоканальных вихретоковых дефектоскопов. Рассматриваются два класса конструктивных элементов рельсового пути: 1) счётчики осей подвижного состава, 2) пересечения рельсовых путей. Длинные отметки, которые не могут быть отнесены к этим двум классам, условно считаются дефектами и выносятся в отдельный третий класс. Для распознавания образов конструктивных элементов на дефектограммах применяется свёрточная нейронная сеть, реализованная в рамках открытой библиотеки TensorFlow. С этой целью каждая выделенная для анализа область дефектограммы преобразуется в графический образ в градации серого цвета размером 30 на 140 точек.

Ключевые слова: неразрушающий контроль рельсов; вихретоковая дефектоскопия; обнаружение дефектов; автоматический анализ дефектограмм; нейронные сети.

ИНФОРМАЦИЯ ОБ АВТОРАХ

Егор Владимирович Кузьмин
автор для корреспонденции

orcid.org/0000-0003-0500-306X.
E-mail: kuzmin@uniyar.ac.ru, kuzminev@nddlab.com
профессор, доктор физ.-мат. наук.

Олег Евгеньевич Горбунов

orcid.org/0000-0001-6274-9971. E-mail: gorbunovoe@nddlab.com
генеральный директор, канд. физ.-мат. наук.

Петр Олегович Плотников

orcid.org/0000-0001-5687-7969. E-mail: plotnikovpo@nddlab.com
инженер-технолог.

Вадим Александрович Тюкин

orcid.org/0000-0001-9149-7435. E-mail: tyukinva@nddlab.com
руков. сектора разработки.

Владимир Анатольевич Башкин

orcid.org/0000-0002-2534-1026. E-mail: bashkinva@nddlab.com
профессор, доктор физ.-мат. наук.

Для цитирования: Е. В. Кузьмин, О. Е. Горбунов, П. О. Плотников, В. А. Тюкин, и В. А. Башкин, "Application of Convolutional Neural Networks for Recognizing Long Structural Elements of Rails in Eddy-Current Defectograms", *Modeling and analysis of information systems*, vol. 27, no. 3, pp. 316-329, 2020.

Введение

Для обеспечения безопасности движения на железнодорожном транспорте регулярно проводится неразрушающий контроль рельсов с применением различных подходов и методов, включая методы вихретоковой дефектоскопии. Актуальной задачей является автоматический анализ [1–3] больших массивов данных (дефектограмм), которые поступают от соответствующего оборудования. Под анализом понимается процесс определения по дефектограммам наличия дефектных участков наряду с выявлением конструктивных элементов рельсового пути.

Данная статья является логическим продолжением работ [4–6], посвященных задаче распознавания образов дефектов и конструктивных элементов железнодорожных рельсов по дефектограммам магнитных и вихретоковых дефектоскопов. Для классификации выделенных конструктивных элементов применяются нейронные сети [7, 8].

В этой работе рассматривается обобщение реального устройства в виде 10-разрядного вихретокового дефектоскопа с 15 каналами данных (на один рельс). Каналы данных соответствуют физическим датчикам, которые последовательно располагаются на поверхности рельса перпендикулярно направлению движения дефектоскопа. Значения амплитуд сигналов каждого канала регистрируются дефектоскопом в виде целых чисел от -512 до 511 (в 10-разрядном случае).

Интерес представляют амплитуды только полезных сигналов. Сигнал считается полезным (и подлежит дальнейшему анализу), если отклонение от нуля как минимум в два раза превосходит пороговый уровень шума рельсов. Отрицательное и положительное пороговые значения уровня шума определяются отдельно для каждого канала данных по алгоритму из статей [5, 6].

Найденные полезные сигналы всех каналов группируются в отметки, которые подлежат дальнейшей классификации. Ранее в статье [4] решалась задача распознавания небольших конструктивных элементов (длиной до 157 мм) следующих трёх типов: 1) болтовой стык с прямым или скошенным соединением рельсов, 2) электроконтактная сварка рельсов и 3) алюминотермитная сварка рельсов.

Данная статья посвящена задаче распознавания записей длинных конструктивных элементов рельсового пути. Выделяются два класса объектов для распознавания: 1) записи мест установки счётчиков осей подвижного состава («axle counter»), 2) записи мест пересечений рельсовых путей («crossing»). Размер записей этих объектов может варьироваться от 420 мм до 3220 мм в длину.

Длинные отметки, которые не могут быть отнесены к этим двум классам, условно считаются дефектами и выносятся в отдельный (третий) класс отметок.

Для распознавания образов конструктивных элементов на вихретоковых дефектограммах была задействована свёрточная нейронная сеть, реализованная в рамках открытой библиотеки TensorFlow (в состав которой входит библиотека Keras) [9]. С этой целью каждая выделенная длинная отметка преобразуется в графический образ в градации серого цвета размером 30 на 140 точек.

Обученная свёрточная нейронная сеть в составе аппаратно-программного комплекса рельсовой дефектоскопии успешно используется на практике для распознавания образов указанных длинных конструктивных элементов рельсового пути.

1. Формирование образа отметки

При автоматическом анализе дефектограммы обычно разбиваются на фрагменты, которые, например, могут соответствовать 50-метровым участкам рельсового пути, т. е. при снятии показаний 15-канального дефектоскопа с каждого миллиметра пути блок анализа представляет собой матрицу размером 15 строк на 50000 столбцов, где элемент матрицы — это значение амплитуды сигнала соответствующего канала данных. Пример графического представления 50-метрового блока анализа приведён на рис. 1 (15-канальные данные с 50-метрового участка одного рельса). В этом фрагменте данных присутствуют записи места пересечения рельсовых путей, места расположения счётчика

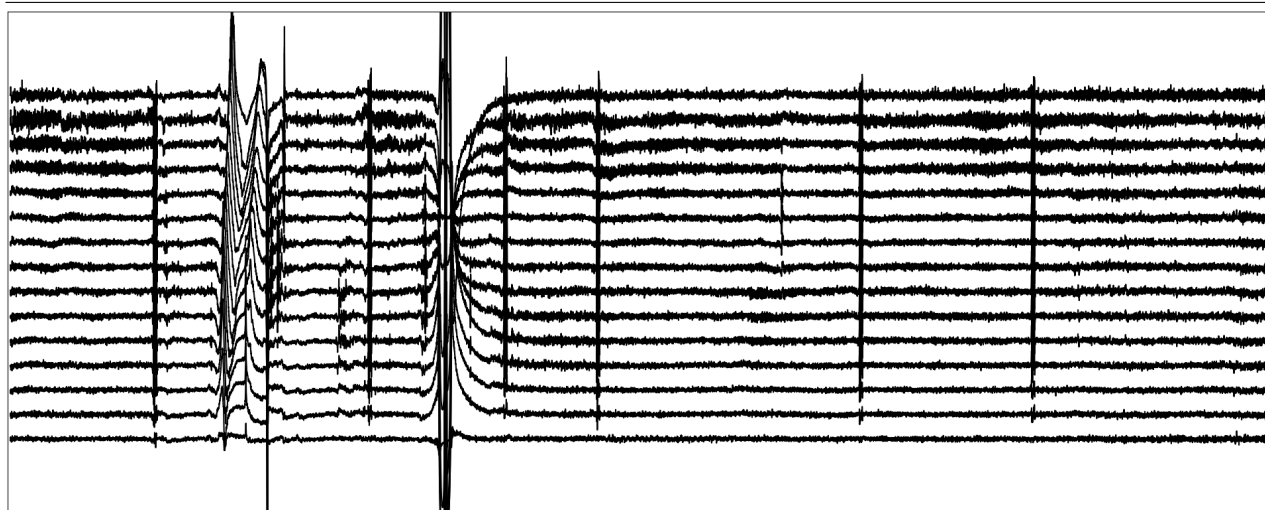


Fig. 1. Eddy-current data (15 channels).
An example of an analysis block (50 m)

Рис. 1. Вихретоковые данные по 15 каналам.
Пример 50-метрового блока анализа

осей подвижного состава, а также записи шести сварных алюминотермитных рельсовых стыков. Более крупно место пересечения путей показано на рис. 2 (см. верхний пример). Типичный вид записи счётчика осей представлен на рис. 3. Запись сварного алюминотермитного стыка можно найти в статье [4] или в несколько искажённом виде (из-за наличия концентрации напряжения в стыке «pumping») на рис. 4.

В данной статье рассматривается задача классификации (распознавания) длинных отметок размером от 420 мм до 3220 мм, которые характеризуются продолжительными участками (от 100 мм) амплитудного отклонения от порогового значения. Наличие таких участков позволяет построчно сжимать графический образ отметки в 3–23 раза без потери её характерного рисунка. Главным образом это касается записей пересечений рельсовых путей, длина которых (и, соответственно, продолжительность участков сильного амплитудного отклонения) зависит как от угла пересечения, так и от сложности рельсовых конструкций.

После того как по полезным сигналам на дефектограмме выделена отметка в виде матрицы амплитуд размером 15 строк (по числу каналов) на n столбцов (длина отметки в мм), предполагается преобразование отметки в графический образ размером 30 на n точек в градации серого цвета с последующим построчным сжатием образа до размера 30 на 140 точек.

Удвоение количества строк в образе отметки. Первым шагом в изменении представления отметки является добавление в матрицу амплитуд по одной строке для каждого канала. При этом в нечетной строке остаются только те амплитуды, которые больше нуля и превосходят положительный пороговый уровень полезных сигналов. Остальные значения в нечетной строке устанавливаются в ноль. В четной строке остаются абсолютные значения только тех амплитуд, которые меньше отрицательного порогового уровня полезных сигналов. Остальные значения в четной строке также устанавливаются в ноль. В результате имеем матрицу размером 30 строк на n столбцов амплитуд полезных сигналов с их дифференциацией и последующим построчным размещением по направлению отклонения от нуля.

Переход к 8-битовым значениям. Следующим шагом является представление текущего образа отметки в виде картинки в градации серого цвета. Для этого необходимо перейти к 8-битовым значениям матрицы. При этом осуществляется перевод амплитуд сигналов в децибелы с абсолютным значением соответствующего уровня полезных сигналов в качестве опорной/исходной величины.

Полученное значение проецируется на шкалу 0–255 простым умножением на 10 и присваиванием 255 всем элементам, которые превосходят это число.

Построчное сжатие. Заключительный шаг — построчное сжатие графического образа отметки до размера 30 строк на 140 столбцов.

Ниже приведена функция на языке Python 3, формирующая графический образ отметки в виде bmp-матрицы размером 30 строк на 140 столбцов в градации серого цвета. В этой функции EC[0:49999,0:14] — это матрица анализируемых исходных вихретоковых данных, т. е. фрагмент дефектограммы длиной 50 метров с данными от 15 каналов, Threshold[0:14] — список положительных и отрицательных пороговых значений амплитуд полезных сигналов для 15 каналов, start и end — координаты начала и конца текущей отметки, B[0:29,0:139] — искомая (изначально состоящая полностью из нулей) bmp-матрица размером 30 строк на 140 столбцов в градации серого цвета, которая возвращается в качестве результата работы функции.

```
def Image_Preparation(EC, Threshold, B, start, end):
    lngth = end - start + 1
    divrslt = lngth / 140
    A = np.zeros((30, lngth))
    for j in range(0, 15):
        thplus = Threshold[j].thplus
        thmins = Threshold[j].thmins
        n = int(0)
        for i in range(start, end + 1):
            ec = EC[i, j]
            if ec > thplus:
                rslt = int(10 * 20 * math.log10(ec / thplus) + 0.5)
                rslt = max(min(rslt, 255), 1)
                A[j * 2, n] = rslt
            if ec < thmins:
                rslt = int(10 * 20 * math.log10(ec / thmins) + 0.5)
                rslt = max(min(rslt, 255), 1)
                A[j * 2 + 1, n] = rslt
            n += 1
    for j in range(0, 30):
        for i in range(0, 140):
            mx = 0
            for m in range(int(i * divrslt), int((i + 1) * divrslt)):
                if mx < A[j, m]:
                    mx = A[j, m]
            B[j, i, 0] = mx
```

На рис. 2 и 3 представлены примеры специального графического изображения в виде bmp-матриц 30 строк на 140 столбцов в градации серого цвета для вихретоковых записей мест пересечения рельсовых путей («crossing») размером 2435 мм и 1361 мм и счётчиков осей («axle counter») размером 2206 мм и 3220 мм соответственно.

Пример условного дефекта — концентрация напряжения в области сварного алюминотермитного рельсового стыка («rumping»). Специальное графическое представление записей этого вида условного дефекта размером 1243 мм и 1323 мм показано на рис. 4.

На этих рисунках в графических образах отметок интенсивность серого цвета возрастает с увеличением значения элемента bmp-матрицы, т. е. от 0 (чистый белый цвет) до 255 (полный черный цвет).

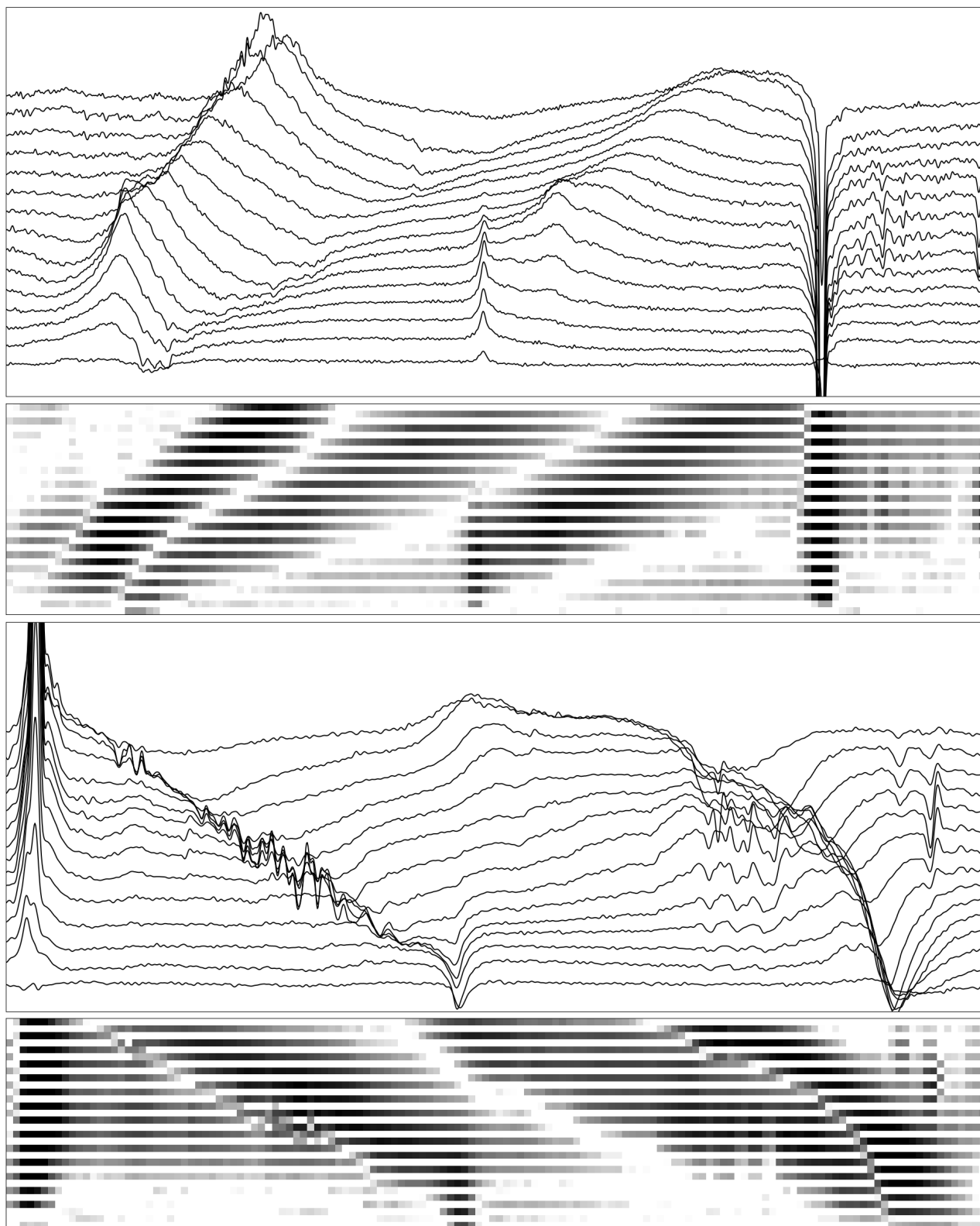


Fig. 2. Eddy-current records and its special graphic representation for crossings

Рис. 2. Вихретоковые записи пересечений путей и их специальное графическое представление

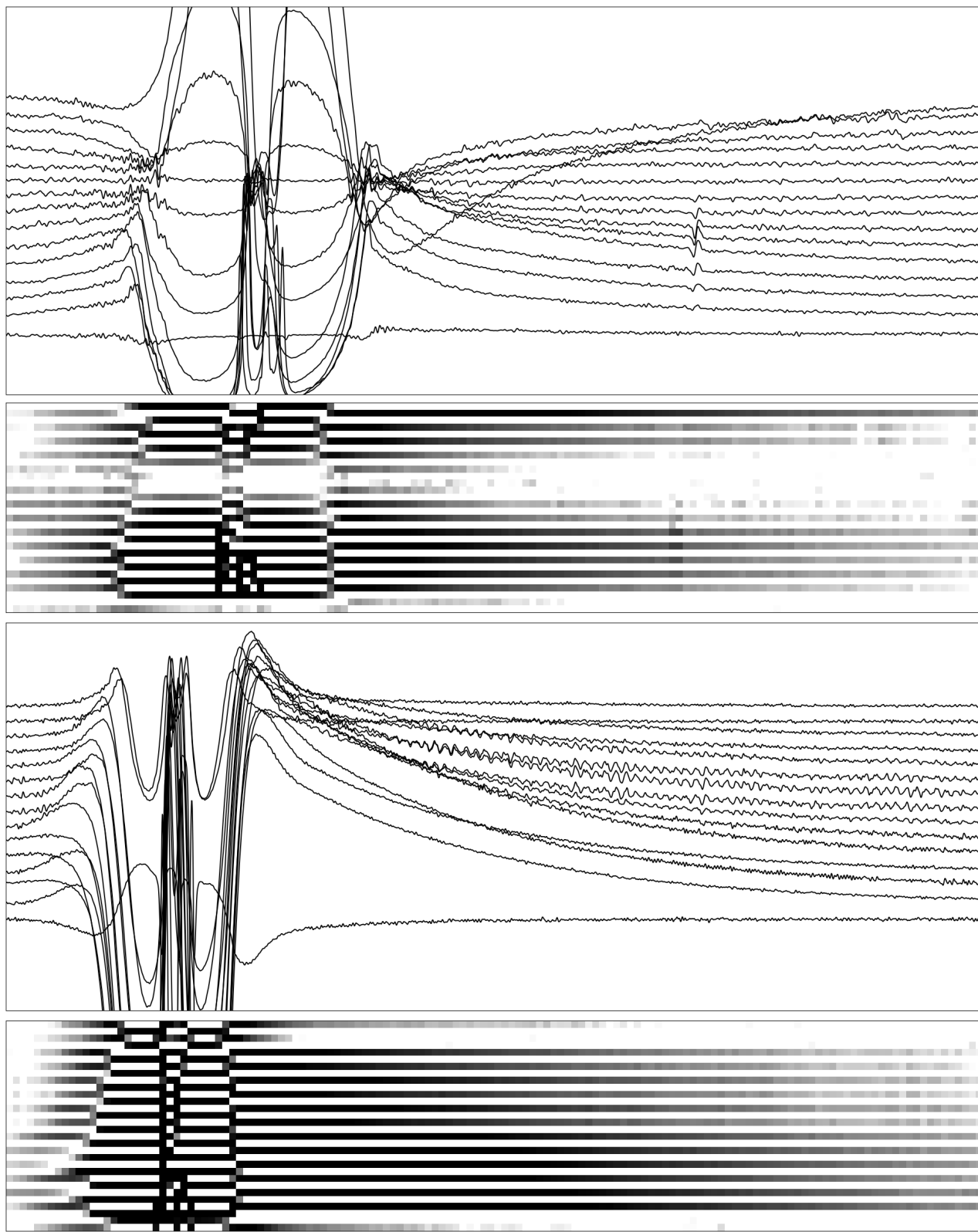


Fig. 3. Eddy-current records and its special graphic representation for axle counters

Рис. 3. Вихретоковые записи счётчиков осей и их специальное графическое представление

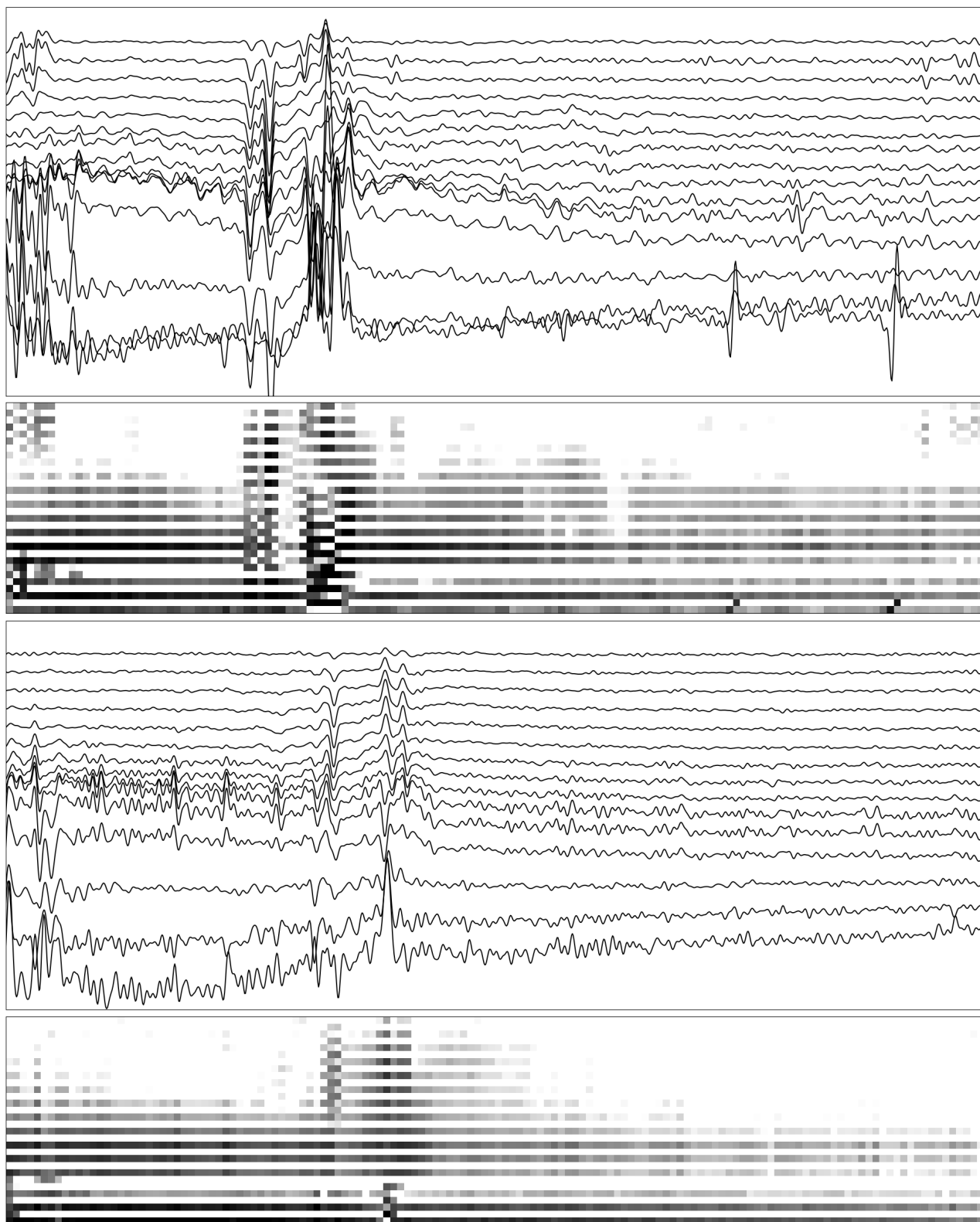


Fig. 4. Eddy-current records and its special graphic representation for pumps

Рис. 4. Вихретоковые записи условных дефектов и их специальное графическое представление

2. Построение и применение нейронной сети

Подготовка данных для обучения нейронной сети. Для обучения нейронной сети был подготовлен набор данных, состоящий из 3720 графических образов (файлов в формате bmp) размером 30 строк на 140 столбцов в градации серого цвета. Этот набор данных содержит в себе следующие три выборки, каждая из которых разбита на три класса — два класса файлов для конструктивных элементов и один для условных дефектов. Обучающая выборка (training set) используется непосредственно для обучения нейронной сети. С помощью проверочной выборки (validation set) в процессе обучения сети осуществляется оценка его качества. Итоговая оценка качества обучения сети после завершения процесса обучения проводится на тестовой выборке (test set).

В подготовленном наборе данных наряду с bmp-файлами, которые были созданы по реальным образам конструктивных элементов и условных дефектов, представленных на вихретоковых дефектограммах, содержатся также и искусственные образцы, сгенерированные на основе реальных по принципу зеркальной симметрии. При построении искусственного образца в графическом образе реальной отметки (bmp-матрице) элементы каждой строки переписываются в обратном порядке.

В обучающей выборке находятся 1860 реальных и 930 сгенерированных образцов условных дефектов и конструктивных элементов двух классов. Проверочная выборка содержит 463 сгенерированных образца. Тестовая выборка составлена из 467 сгенерированных образцов. Таким образом, проверочная и тестовые выборки в совокупности занимают 25% от общего числа подготовленных для обучения графических файлов. А все входящие в них образцы конструктивных элементов являются сгенерированными. Наличие в обучающей выборке всех реальных образцов связано с необходимостью «запоминания» нейронной сетью всех подтвержденных экспертами образов конструктивных элементов, которые были найдены на реальных вихретоковых дефектограммах.

Отметим, что указанное количество графических образцов значительно меньше того объёма данных, который использовался в работе [4] при обучении нейронной сети для распознавания сварок и рельсовых стыков. Это связано с тем, что на вихретоковых дефектограммах записи счётчиков осей и элементов стрелочных переводов встречаются намного реже, чем записи рельсовых стыков и алюминотермитных и электроконтактных сварок. Однако даже такое относительно небольшое число образцов позволило обучить нейронную сеть, успешно применяемую на практике.

Создание нейронной сети. Следующий код на языке Python 3 с использованием библиотек TensorFlow и Keras позволяет сформировать свёрточную нейронную сеть, состоящую из чередующихся двух слоёв подвыборки и двух слоёв свёртки, полносвязного внутреннего слоя из 32 нейронов, слоя регуляризации и полносвязного выходного слоя, состоящего всего лишь из 3 нейронов (по числу классов распознавания).

Начальные значения весовых коэффициентов нейронов назначаются случайным образом в соответствии с нормальным распределением вероятностей. Функция активации нейронов внутренних слоёв — ReLu (Rectified Linear Unit $f(x) = \max(0, x)$). В качестве функции активации нейронов выходного слоя используется нормализованная экспоненциальная функция SoftMax, при которой сумма значений всех выходных нейронов равняется 1. Эта функция позволяет выдавать результат в виде набора вероятностей принадлежности распознаваемого объекта тому или иному классу (искомый класс определяется по большему значению из набора).

Слой регуляризации служит для борьбы с проблемой переобучения сети (адаптации весов только к обучающей выборке), который выключает нейроны внутреннего слоя из процесса обучения с вероятностью 0,25 (на один нейрон) при каждом новом объекте, подаваемом на сеть для обучения.

Метод обучения сети — метод стохастического градиентного спуска в модификации Adam. Мера ошибки при обучении — перекрестная энтропия (для трёх категорий). Метрика оценки результатов обучения — точность (ассигасу, отношение количества правильно классифицированных объектов к общему числу классифицируемых объектов).

Построенная свёрточная нейронная сеть имеет относительно небольшой объём — во время обучения настраиваются 52139 параметров (обучаемых весов).

```
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Conv2D, MaxPooling2D
from tensorflow.python.keras.layers import Activation, Dropout, Flatten, Dense
img_width, img_height = 140, 30 # размеры изображения
input_shape = (img_height, img_width, 1) # размерность тензора
model = Sequential() # создание последовательной модели нейронной сети
# добавление слоя свёртки с размером ядра 3x3; 16 карт признаков
model.add(Conv2D(16, (3, 3), input_shape=input_shape))
model.add(Activation('relu')) # функция активации ReLu
# добавление слоя подвыборки; выбор максимального значения из квадрата 2x2
model.add(MaxPooling2D(pool_size=(2, 2)))
# добавление слоя свёртки с размером ядра 3x3; 8 карт признаков
model.add(Conv2D(8, (3, 3)))
model.add(Activation('relu')) # функция активации ReLu
# добавление слоя подвыборки; выбор максимального значения из квадрата 2x2
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten()) # добавление полносвязного внутреннего слоя,
model.add(Dense(32)) # состоящего из 32-х нейронов
model.add(Activation('relu')) # функция активации ReLu
model.add(Dropout(0.25)) # добавление слоя регуляризации
# добавление выходного слоя из трёх нейронов
model.add(Dense(3, activation="softmax", kernel_initializer="normal"))
# компиляция модели
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Обучение нейронной сети. Ниже приводится код на языке Python 3, с помощью которого происходит обучение нейронной сети для распознавания конструктивных элементов и условных дефектов. Данные для обучения, проверки и тестирования подаются порциями по 32 изображения из соответствующих каталогов (для этого задействованы генераторы изображений библиотеки Keras с размером мини-выборки, равным 32). При этом осуществляется нормализация данных — значения элементов bmp-матриц делятся на 255 (приводятся к вещественным числам от 0 до 1). Количество эпох обучения — 400.

Время, затраченное на обучение описанной модели нейронной сети, составило около 30 минут. Обучение сети проводилось на компьютере с центральным процессором Intel Core i7-4770K CPU 3.50 GHz и оперативной памятью 8 GB. В результате обучения была достигнута следующая точность (accuracy):

- на данных для обучения — 100%,
- на данных для проверки — 96,88%,
- на тестовых данных — 96,43%.

```
from tensorflow.python.keras.preprocessing.image import ImageDataGenerator
epochs = 400 # количество эпох обучения
batch_size = 32 # размер мини-выборки
nb_train_samples = (1014 + 822 + 954) # количество изображений для обучения
nb_validation_samples = (168 + 136 + 159) # количество изображений для проверки
nb_test_samples = (170 + 138 + 159) # количество изображений для тестирования
```

```

# создание генератора изображений с нормализацией данных
datagen = ImageDataGenerator(rescale=1. / 255)
# генерация данных для обучения (на основе изображений из каталога)
train_generator = datagen.flow_from_directory(
    train_dir, # каталог с данными для обучения
    target_size=(img_height, img_width), # размер изображения
    color_mode='grayscale', # изображение в градации серого цвета
    batch_size=batch_size, # размер мини-выборки
    class_mode='categorical')
# генерация данных для проверки (на основе изображений из каталога)
val_generator = datagen.flow_from_directory(
    val_dir, # каталог с данными для проверки
    target_size=(img_height, img_width), # размер изображения
    color_mode='grayscale', # изображение в градации серого цвета
    batch_size=batch_size, # размер мини-выборки
    class_mode='categorical')
# генерация данных для тестирования (на основе изображений из каталога)
test_generator = datagen.flow_from_directory(
    test_dir, # каталог с данными для тестирования
    target_size=(img_height, img_width), # размер изображения
    color_mode='grayscale', # изображение в градации серого цвета
    batch_size=batch_size, # размер мини-выборки
    class_mode='categorical')
# обучение модели нейронной сети с помощью генераторов данных
model.fit_generator(
    train_generator,
    steps_per_epoch=nb_train_samples // batch_size,
    epochs=epochs, # количество эпох обучения
    validation_data=val_generator,
    validation_steps=nb_validation_samples // batch_size)
# оценка качества обучения на тестовых данных
scores = model.evaluate_generator(test_generator, nb_test_samples // batch_size)

```

Распознавание графических образов конструктивных элементов. Отметим, что в каждом из каталогов `train_dir`, `val_dir` и `test_dir` данные (картинки) трёх классов находились в соответствующих каталогах: для графических образов счётчиков осей был отведен каталог `Axle`, для образов мест пересечения рельсовых путей — каталог `Cross`, а для условных дефектов — `Pump`. Поэтому при обучении нейронной сети классы образцов получили имена в соответствии с названиями этих каталогов.

С применением следующего кода на языке Python 3 происходит распознавание графических образов трёх классов с помощью обученной нейронной сети. На вход сети подается подготовленный и нормализованный графический образ B найденной на дефектограмме отметки, выходом является название класса, к которому принадлежит исследуемый образ. Нейронная сеть выдает набор значений вероятностей принадлежности объекта к соответствующему классу. Класс объекта выбирается по максимальному значению из этого набора.

```

import numpy
classes = ['axle', 'cross', 'pump']
B /= 255 # нормализация матрицы B
B = numpy.expand_dims(B, axis=0) # вспомогательное увеличение размерности

```

```

preds = model.predict(B) # классификация входного объекта
if classes[preds.argmax()] == 'axle': # расшифровка результата классификации
    mark_result = 'axle_counter' # счётчик осей
if classes[preds.argmax()] == 'cross': # расшифровка результата классификации
    mark_result = 'crossing' # место пересечения путей
if classes[preds.argmax()] == 'pump': # расшифровка результата классификации
    mark_result = 'pumping' # условный дефект

```

На рис. 5 и 6 изображены карты признаков, которые были получены на соответствующих свёрточных слоях нейронной сети для графического образа места пересечения рельсовых путей, приведённого на рис. 2 (см. верхний пример) (для наглядности максимальное значение точки соответствует чистому белому цвету, а минимальное — полному чёрному; размер карт первого слоя составляет 28x138 точек, второго слоя — 12x67). Карты признаков для графического образа счётчика осей (рис. 3, верхний пример) показаны на рис. 7 и 8.

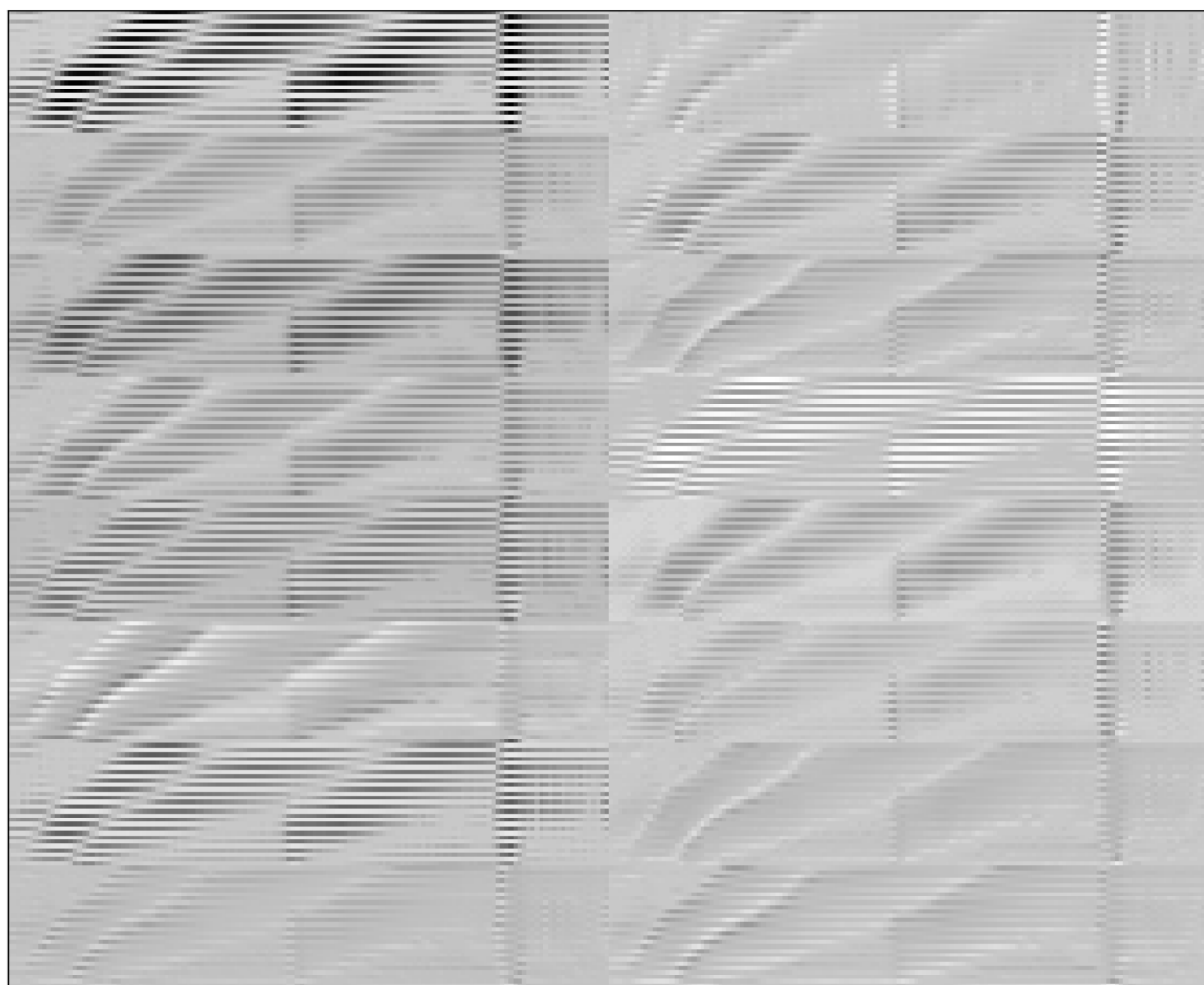


Fig. 5. Feature maps for the “crossing” in the first convolutional layer of the neural network

Рис. 5. Карты признаков для «пересечения» на первом свёрточном слое нейронной сети



Fig. 6. Feature maps for the “crossing” in the second convolutional layer of the neural network

Рис. 6. Карты признаков для «пересечения» на втором свёрточном слое нейронной сети

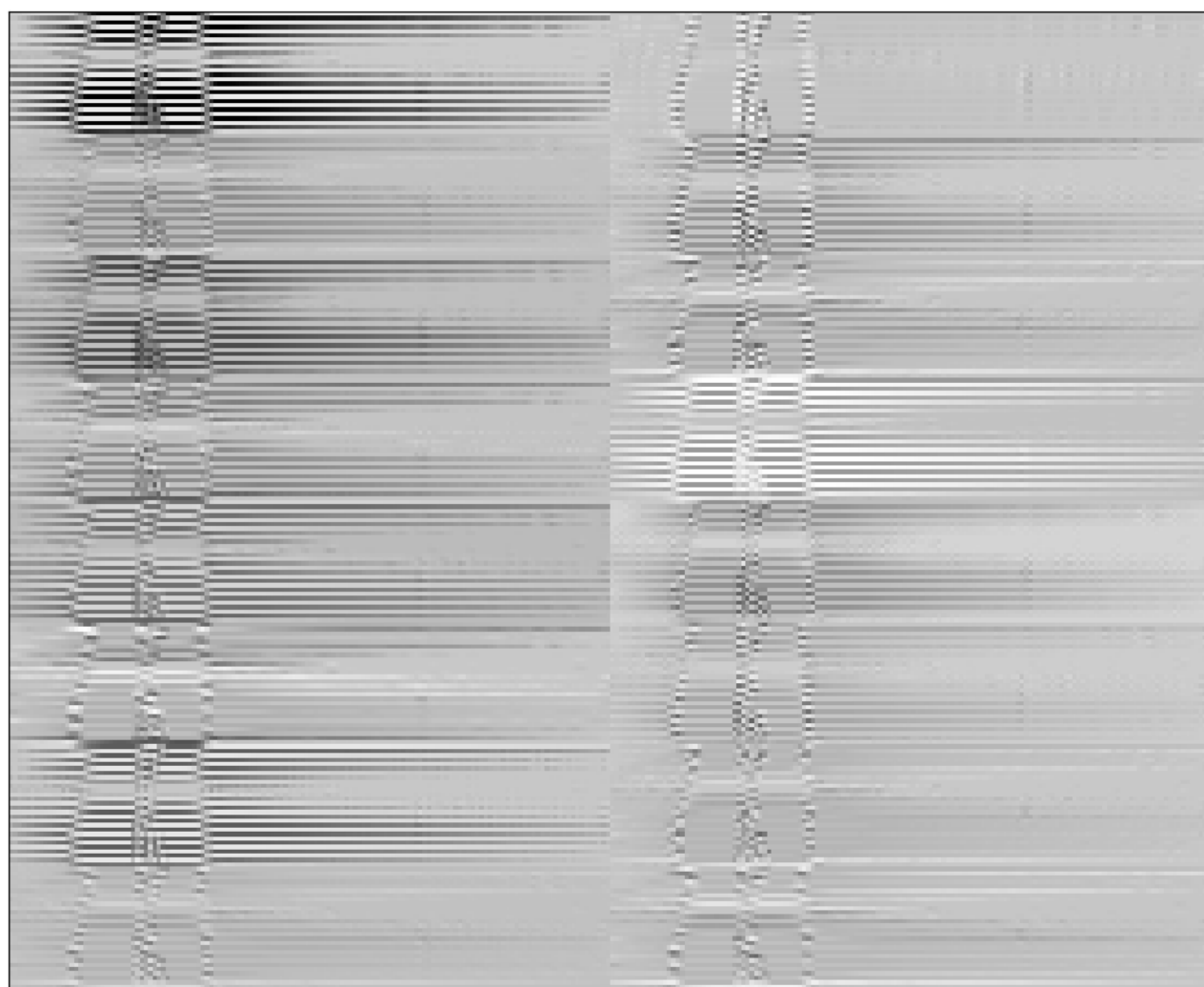


Fig. 7. Feature maps for the “axle counter” in the first convolutional layer of the neural network

Рис. 7. Карты признаков для «счётчика осей» на первом свёрточном слое нейронной сети

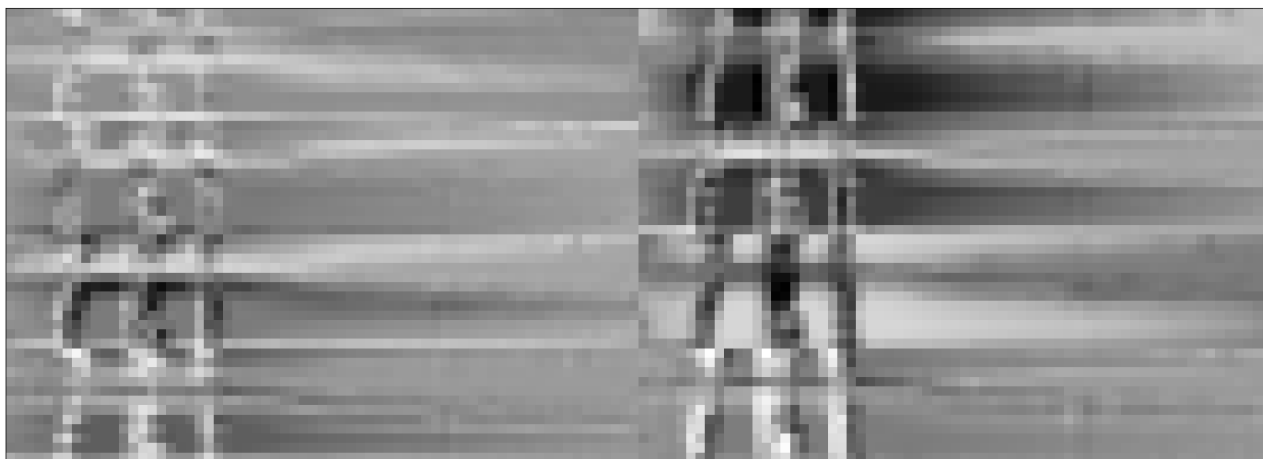


Fig. 8. Feature maps for the “axle counter” in the second convolutional layer of the neural network

Рис. 8. Карты признаков для «счётчика осей» на втором свёрточном слое нейронной сети

Заключение

По результатам обучения можно сделать вывод, что выбранная конфигурация модели нейронной сети является удачной и вполне достаточной для решения поставленной задачи распознавания длинных конструктивных элементов рельсового пути при неразрушающем контроле поверхности головки рельсов с помощью вихретоковых дефектоскопов.

References

- [1] A. A. Markov and E. A. Kuznetsova, *Rails flaw detection. Formation and analysis of signals. Book 1. Principles*. St. Petersburg: KultInformPress, 2010.
- [2] A. A. Markov and E. A. Kuznetsova, *Rails flaw detection. Formation and analysis of signals. Book 2. Data interpretation*. St. Petersburg: Ultra Print, 2014.
- [3] V. F. Tarabrin, A. V. Zverev, O. E. Gorbunov, and E. V. Kuzmin, “About Data Filtration of the Defectogram Automatic Interpretation by Hardware and Software Complex ASTRA”, *NDT World*, vol. 64, no. 2, pp. 5–9, 2014.
- [4] E. V. Kuzmin, O. E. Gorbunov, P. O. Plotnikov, V. A. Tyukin, and V. A. Bashkin, “Application of Neural Networks for Recognizing Rail Structural Elements in Magnetic and Eddy Current Defectograms”, *Automatic Control and Computer Sciences*, vol. 53, no. 7, pp. 628–637, 2019.
- [5] E. V. Kuzmin, O. E. Gorbunov, P. O. Plotnikov, and V. A. Tyukin, “An Efficient Algorithm for Finding the Level of Useful Signals on Interpretation of Magnetic and Eddy Current Defectograms”, *Automatic Control and Computer Sciences*, vol. 52, no. 7, pp. 867–870, 2018.
- [6] E. V. Kuzmin, O. E. Gorbunov, P. O. Plotnikov, and V. A. Tyukin, “Finding the Level of Useful Signals on Interpretation of Magnetic and Eddy-Current Defectograms”, *Automatic Control and Computer Sciences*, vol. 52, no. 7, pp. 658–666, 2018.
- [7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [8] F. Chollet, *Deep Learning with Python*. Manning Publications Co., 2018.
- [9] *TensorFlow*. [Online]. Available: <https://www.tensorflow.org/>.

Automated Search and Analysis of the Stylometric Features that Describe the Style of the Prose 19th–21st Centuries

K. V. Lagutina¹, A. M. Manakhova¹

DOI: [10.18255/1818-1015-2020-3-330-343](https://doi.org/10.18255/1818-1015-2020-3-330-343)

¹P. G. Demidov Yaroslavl State University, Sovetskaya str., 14, Yaroslavl, 150003, Russia.

MSC2020: 68T50

Research article

Full text in Russian

Received May 14, 2020

After revision June 8, 2020

Accepted June 10, 2020

The article is devoted to comparison of stylometric features of several levels, which are markers of the style of the prose text and analysis of the stylistic changes in Russian and British prose of the 19th–21st centuries. Stylometric features include the low-level features based on the words and symbols and high-level based on rhythmic. These features model the style of a text and are the indicators of the time when the text was created.

Calculations of all the features are performed completely automatically, so it allows to conduct the large-scale experiments with artworks of a large volume and speeds up the work of a linguist. To calculate the stylometric features including ones based on the search results for rhythmic figures the ProseRhythmDetector program is used. As a result of its work, each text is presented as a set of the same features of three levels: characters, words, rhythm. Texts are combined by decades, for each decade there are found average values of stylometric features. The obtained models of decades are compared using standard similarity metrics, results of comparison are visualized in the form of the heat maps and dendrograms. Experiments with two corpora of Russian and British texts show that during the 19th–21st centuries there are general trends in style change for both corpora, for example, a decrease in the number of rhythmic figures per sentence, and also particular trends for each language, for example, dynamics of change of the word and sentence lengths. Stylometric features of all levels reveal the similarity in the style of texts published in one century. Also, features of three levels in the complex better demonstrate the uniqueness of each decade than features of a particular level. This study shows the importance of stylometric features as style markers of the different eras and allows us to identify trends in style during several centuries.

Keywords: text rhythm; rhythm analysis; natural language processing; stylometry; rhythm figures; automation.

INFORMATION ABOUT THE AUTHORS

Ksenia Vladimirovna Lagutina	orcid.org/0000-0002-1742-3240 . E-mail: lagutinakv@mail.ru
correspondence author	postgraduate student.
Alla Mikhajlovna Manakhova	orcid.org/0000-0001-7429-3529 . E-mail: al.mnkhv@yandex.ru
	student.

Funding: The study was funded by RFBR according to the research project №19-07-00243.

For citation: K. V. Lagutina and A. M. Manakhova, “Automated Search and Analysis of the Stylometric Features that Describe the Style of the Prose 19th–21st Centuries”, *Modeling and analysis of information systems*, vol. 27, no. 3, pp. 330–343, 2020.

Автоматизированный поиск и анализ стилометрических характеристик, описывающих стиль прозы 19–21 веков

К. В. Лагутина¹, А. М. Манахова¹

DOI: [10.18255/1818-1015-2020-3-330-343](https://doi.org/10.18255/1818-1015-2020-3-330-343)

¹Ярославский государственный университет им. П. Г. Демидова, ул. Советская, 14, г. Ярославль, 150003 Россия.

УДК 004.912

Научная статья

Полный текст на русском языке

Получена 14 мая 2020 г.

После доработки 8 июня 2020 г.

Принята к публикации 10 июня 2020 г.

Статья посвящена сравнению стилометрических характеристик нескольких уровней, являющихся маркерами стиля прозаического текста, и анализу стилистических изменений русской и британской прозы 19–21 веков. Стилометрические характеристики включают в себя низкоуровневые характеристики, основанные на словах и символах, и высокоуровневые — ритмические. Подобные характеристики моделируют стиль текста и являются индикаторами времени его создания.

Вычисление всех характеристик происходит полностью автоматически, что позволяет проводить крупные эксперименты с художественными произведениями большого объёма и ускоряет работу эксперта-лингвиста. Для подсчёта стилометрических характеристик, в том числе основанных на результатах поиска ритмических средств, используется программа ProseRhythmDetector. В результате её работы каждый текст представляется в виде набора одних и тех же характеристик трёх уровней: символов, слов, ритма. Тексты объединяются по десятилетиям, для каждого десятилетия находятся средние значения стилометрических характеристик. Полученные модели десятилетий сравниваются при помощи стандартных метрик близости, результаты сравнения визуализируются в виде тепловых карт и дендрограмм. Эксперименты с двумя корпусами русских и британских текстов показывают, что в течение 19–21 веков появляются как общие тенденции изменения стиля для обоих корпусов, например, уменьшение количества ритмических средств в расчёте на одно предложение, так и собственные для каждого языка, например, динамика изменения длин слов и предложений. Стилометрические характеристики всех уровней выявляют схожесть стиля текстов, опубликованных в одном веке. Также характеристики трёх уровней в комплексе лучше демонстрируют уникальность каждого десятилетия, чем характеристики конкретного уровня. Это исследование показывает значимость стилометрических характеристик как маркеров стиля различных эпох и позволяет выявить тенденции изменения стиля на протяжении нескольких веков.

Ключевые слова: ритм текста; анализ ритма; обработка естественного языка; стилометрия; ритмические средства; автоматизация.

ИНФОРМАЦИЯ ОБ АВТОРАХ

Ксения Владимировна Лагутина
автор для корреспонденции
Алла Михайловна Манахова

orcid.org/0000-0002-1742-3240. E-mail: lagutinakv@mail.ru

аспирант.

orcid.org/0000-0001-7429-3529. E-mail: al.mnhv@yandex.ru

студент.

Финансирование: Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта №19-07-00243.

Для цитирования: K. V. Lagutina and A. M. Manakhova, “Automated Search and Analysis of the Stylometric Features that Describe the Style of the Prose 19th–21st Centuries”, *Modeling and analysis of information systems*, vol. 27, no. 3, pp. 330–343, 2020.

Введение

Ритм текста определяется как регулярное повторение схожих единиц речи [1]. В литературе отмечается, что ритм прозы отличается от ритма поэзии и требует собственных методик определения, в том числе таких, которые бы моделировали его в количественном виде и позволяли бы сравнивать прозаические тексты между собой [2].

Количественные характеристики ритма текста входят в обширную группу стилометрических характеристик, которые на разных уровнях описывают стиль текста и составляют его числовую модель [3], на основе которой можно проводить статистический и сравнительный анализ текстов различных авторов, жанров, временных эпох и т.п.

Несмотря на широкую распространённость стилометрических характеристик в сфере обработки естественного языка, они, как правило, используются для решения конкретных задач классификации, например, определения автора или жанра. Более того, из всего спектра характеристик наиболее изученными являются низкоуровневые, которые моделируют текст на уровне слов или символов. Высокоуровневые или лингвистические характеристики, включая ритмические, являются малоизученной частью стиля текста [4].

Авторы поставили перед собой задачу сравнить высокоуровневые и низкоуровневые стилометрические характеристики в прозаических текстах 19–21 веков по десятилетиям и статистически проанализировать динамику изменения стиля русской и британской прозы. Для этого была использована программа ProseRhythmDetector, которая была представлена в предыдущих исследованиях как инструмент для поиска ритмических средств в прозаических текстах. Авторы добавили в ProseRhythmDetector модуль, который вычисляет стилометрические характеристики текста уровня символов, слов и ритма. Данная статья описывает эксперименты с этим модулем по анализу текстов по векам и десятилетиям.

1. Обзор смежных работ

Стилометрия – научная дисциплина, занимающаяся измерением стилизованных характеристик текстов с целью их упорядочивания, диагностики, идентификации, параметризации, таксономии, атрибуции и периодизации [5]. Стилометрические характеристики прозаических текстов изменяются с течением времени для литературы на разных языках, поэтому они могут служить индикаторами эпохи создания произведений [6]. Кумар и другие исследуют лексические маркеры стиля для 18–21 веков, по которым можно определить дату публикации. Используя характеристики текста уровня слов и фраз, они достигают средней ошибки в 32 года, т.е. достаточно точно определяют век публикации текста.

Для решения подобных задач обычно берутся простые характеристики уровня слов и символов. Авторы статьи [7] строили именно такую модель, но по результатам исследования рекомендовали расширять модели текстов более сложными лингвистическими характеристиками.

Контекст Semeval 2015 [8] был посвящён определению временного периода публикации статьи в диапазоне от 18 до 21 века. Лучших результатов до 86.8 % точности достигли участники, которые применили широкий спектр стилометрических характеристик: от лексических до грамматических, включая даже мета-свойства документа.

Гопиди и Алам [9] показали, что числовые грамматические характеристики, а также характеристики, основанные на ударениях и рифме, отличаются для разных 50-летий для прозы и поэзии.

Эти результаты доказывают, что стилометрические характеристики различных уровней хорошо моделируют стиль текста и могут указывать на конкретную эпоху его создания.

Стилометрические характеристики отличаются не только для разных эпох, но и для разных языков. Авторы работы [10] кластеризовали тексты при помощи алгоритма k -средних, основываясь на встречаемости слов и символов. F -мера для такого алгоритма классификации получилась не выше

53 %. С применением нейросетей классификация текстов по языкам на основе характеристик уровня символов и слов может достигать более высоких значений F -меры 70–80 % [11]. Но такие исследования затрагивают только низкоуровневые характеристики, оставляя открытым вопрос значимости лингвистических характеристик для моделирования и анализа стиля.

Авторы в предыдущей работе [12] исследовали вариативность ритмических характеристик для различных периодов времени (19–21 веков) и языков (русского и английского), где показали, что все три века отличаются по ритму. В этой статье описываются результаты сравнительного анализа стилометрических характеристик нескольких уровней.

2. Стилиметрические характеристики

Стилиметрический анализ текста включает в себя поиск и подсчёт различных стилометрических характеристик. Среди таких характеристик можно выделить несколько категорий:

1. Уровень символов;
2. Уровень слов;
3. Уровень ритма.

Ритмические характеристики текста определяются исходя из употребления средств создания ритма, в основе которых находится повтор в определенной конфигурации, в определенной позиции, с определенным количеством повторяющихся элементов. Для данной работы были выбраны следующие ритмические средства:

1. Анафора — скрепление речевых отрезков (частей фразы, стихов) с помощью повтора слова или словосочетания в начальной позиции.
2. Эпифора — скрепление речевых отрезков (частей фразы, стихов) с помощью повтора слова или словосочетания в конечной позиции.
3. Симплока — фигура синтаксического параллелизма в смежных стихах или фразах, у которых одинаковые начало и конец при разной середине или наоборот, разные начало и конец при одинаковой середине.
4. Анадиплозис — риторическая фигура, в которой следующее предложение начинается теми же словами, которыми оканчивается предыдущее.
5. Эпаналепсис — фигура речи, состоящая в повторении одного и того же слова или словосочетания с небольшими вариациями.
6. Многосоюзие — стилистическая фигура, состоящая в намеренном увеличении количества союзов в предложении, обычно для связи однородных членов.
7. Диакоса — риторический термин для повторения слова или фразы, разбитых на одно или несколько промежуточных слов
8. Эпизевкис — фигура речи, которая обозначает повторение слов без разрыва между повторениями.

Для данных ритмических средств были выбраны следующие числовые стилометрические характеристики:

1. Количество появлений в тексте конкретного средства, делённое на количество предложений;
2. Количество появлений в тексте всех средств, делённое на количество предложений;
3. Доля уникальных слов среди всех, составляющих средства, в данном случае тех, которые повторяются только один раз;
4. Доли существительных, прилагательных, глаголов и наречий среди слов, составляющих средства.

Выбор данных средств для анализа ритма, а именно для их автоматизированного поиска и количественной обработки обусловлен тем, что это наиболее частотные ритмические средства, употребляемые в прозаических текстах. Именно они выделяются в качестве ритмических средств на лексико-грамматическом уровне большинством лингвистов, проводящих исследования в области ритмизации текста [12].

В качестве стилометрических характеристик на уровне символов и слов были выбраны нижеперечисленные характеристики.

На уровне символов:

1. Количество букв, как отдельных, так и их общее количество;
2. Количество символов, как отдельных, так и их общее количество;
3. Средняя длина предложения в символах.

На уровне слов:

1. Количество слов;
2. Количество предложений;
3. Средняя длина предложений по количеству слов;
4. Средняя длина слова.

Выбор данных стилометрических средств на уровне символов и слов был обусловлен тем, что они являются наиболее показательными при определении авторского стиля во время исследования произведения [4].

3. Постановка экспериментов

3.1. Основные этапы экспериментов

Стилометрические характеристики трёх разных уровней вычисляются и визуализируются автоматически. Эксперименты с этими характеристиками были поставлены следующим образом:

- Сначала в текстах были выявлены ритмические средства. Алгоритмы поиска ритмических средств были взяты из статьи [12].
- Для выявленных ритмических средств были подсчитаны стилометрические характеристики.
- Параллельно с подсчётом характеристик ритма для текстов были вычислены стилометрические характеристики уровня слов и символов.
- Стилометрические характеристики текстов были агрегированы по десятилетиям, десятилетия сравнивались между собой.
- На последнем этапе результаты сравнения были визуализированы с помощью тепловых карт и иерархической кластеризации.

На первом этапе ритмические средства выявляются с точностью 80–95 %. Таким образом авторы получают качественную модель ритма текста.

Ритмические и простые стилометрические характеристики вычисляются на основе текста и модели его ритма по точным правилам, описанным в предыдущем разделе. В результате каждый текст представляется как вектор числовых характеристик. Вектора сравниваются при помощи мер близости, на основе которых организована визуализация результатов.

3.2. Визуализация стилометрических характеристик

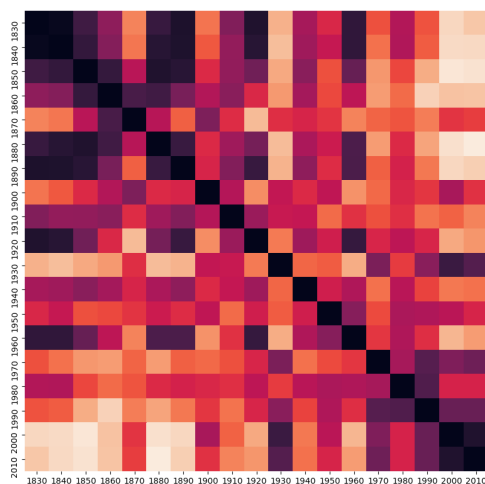
После того как вычислены стилометрические характеристики для текстов, считаются стилометрические характеристики для десятилетий. Для каждого десятилетия берутся средние значения характеристик текстов, опубликованных в этот период. Таким образом получаются вектора характеристик десятилетий такого же типа, как и вектора для отдельных текстов.

Стилометрические характеристики десятилетий и их сравнение визуализируются тремя способами:

- В виде тепловых карт, которые описывают близость десятилетий по стилю. Это квадратные тепловые карты, на осях которых расположены десятилетия, а оттенок в ячейке обозначает степень близости пары десятилетий: чем темнее оттенок, тем ближе объекты друг к другу. В качестве меры близости использовались четыре популярные метрики: расстояние Чебышёва, коэффициент корреляции, расстояние Евклида и манхэттенское расстояние.
- В виде тепловых карт, которые описывают диапазоны значений стилометрических характеристик. На горизонтальной оси располагаются названия конкретных характеристик, на вертикальной — десятилетия. Ячейки карты содержат значение характеристики, а также имеют цвет, оттенок которого обозначает величину значения относительно других. Самые большие значения обозначаются светлыми оттенками, самые маленькие — тёмными. Справа на карте отображается столбик с диапазоном значений и оттенками для разных значений.
- В виде дендрограмм, полученных в результате кластеризации. Листья дендрограммы — это десятилетия, они размещены по горизонтали. По вертикали отмечаются расстояния между кластерами в виде горизонтальных отрезков на определённом уровне. Дендрограмма строится с помощью агломеративного подхода, от листьев к стволу. Метрики близости используются те же, что и для тепловых карт. В качестве функций расстояния между кластерами применяются три метода: одиночной, средней и полной связи.

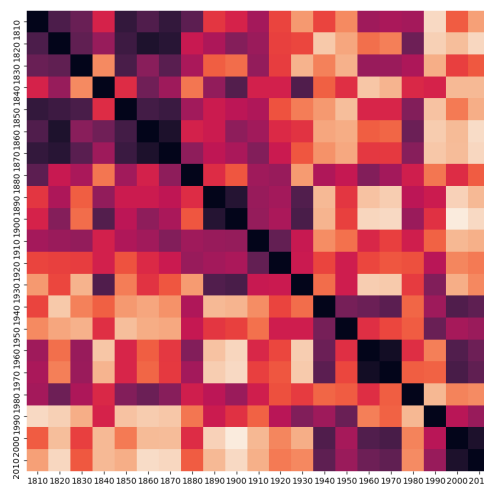
Для сравнения десятилетий стилометрические характеристики предварительно нормализуются: из конкретного значения вычитается среднее значение данной характеристики по всему корпусу текстов, полученная разность делится на среднеквадратическое отклонение данной характеристики.

Все три способа визуализации достаточно наглядны и позволяют проанализировать как динамику изменения стилометрических характеристик на протяжении десятилетий, так и близость десятилетий друг к другу с точки зрения стиля.



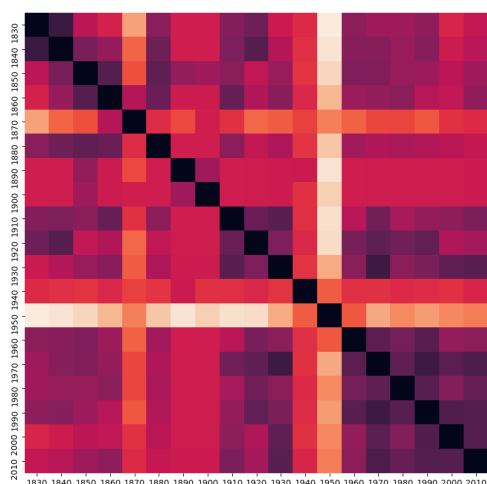
a)

Fig. 1. All levels, correlation metric, a) Russian, b) English

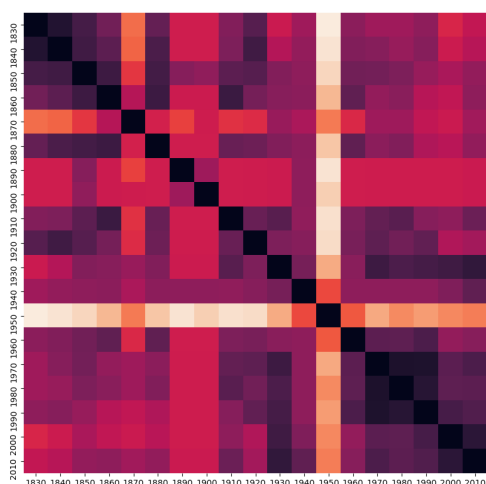


b)

Рис. 1. Все уровни вместе, метрика корреляции, а) русский язык, б) английский язык



a)
Fig. 2. a) All levels, b) symbol level, Chebyshev
metric, Russian



b)
Рис. 2. а) Все уровни вместе, б) уровень
символов, метрика Чебышёва, русский язык

4. Эксперименты

4.1. Программная реализация и корпус

Инструмент *ProseRhythmDetector* ¹, позволяющий выявлять и подсчитывать стилометрические характеристики, был разработан на языке Python. При разработке также использовалась библиотека *textblob*, которая была особенно полезна при подсчёте слов и предложений в тексте.

После завершения разработки был проведен ряд экспериментов на основе двух корпусов текстов. Один из них на английском языке, а другой — на русском. Каждый корпус включает в себя по 243 произведения более 90 известных авторов. У каждого из текстов указана дата публикации: для текстов на английском языке с 1815 по 2019 гг., а для текстов на русском языке с 1832 по 2019 гг. Каждый текст содержит в себе до 425 000 слов.

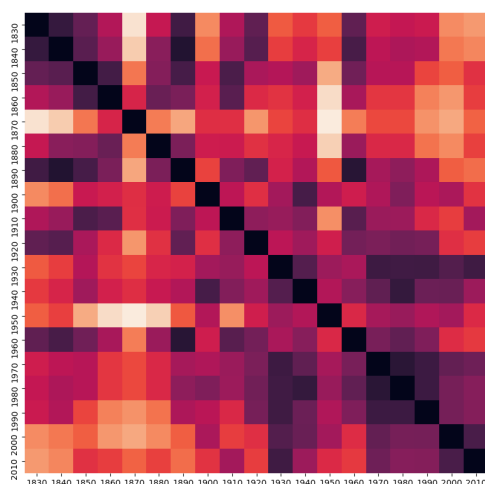
4.2. Тепловые карты близости

На основе получившихся векторов характеристик по корпусу текстов были построены 4 набора тепловых карт по метрикам близости Чебышёва, Евклида, корреляции и манхэттенского расстояния. Метрики Евклида и манхэттенское расстояние показали практически идентичные результаты на всех рассматриваемых уровнях. Относительно заметные различия наблюдаются только при объединении всех уровней.

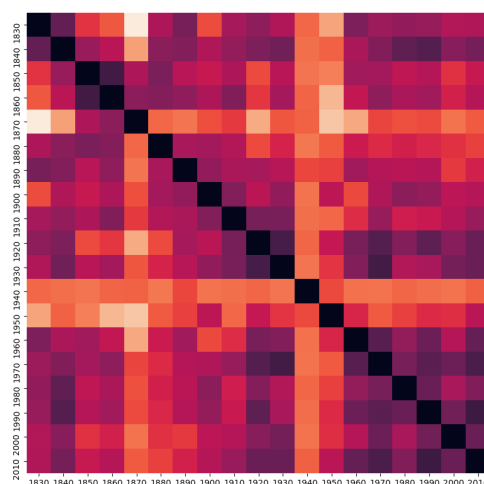
Метрика корреляции не сильно помогла в анализе литературы за указанный период, однако она продемонстрировала близость 00-х и 10-х годов 21-ого века на обоих корпусах (см. Рис. 1). Кроме того на тепловой карте, построенной на основе корпуса русских текстов, можно отметить особую близость 1830-х и 1840-х годов. Также по карте видно, что 21 век более далёк от 19-го века, чем от 20-го, а ещё можно увидеть, что 1960-е и 1920-е годы близки к началу 19-го столетия. Наконец, по карте можно сделать вывод, что литература конца 19-го века была весьма похожа на литературу начала 19-го века, в то время как произведения середины 19-го столетия сильно отличаются от них.

По тепловой карте, построенной на основе корпуса английских текстов с расчётом по метрике корреляции можно понять, что произведения большей части 19-го века близки между собой. Но,

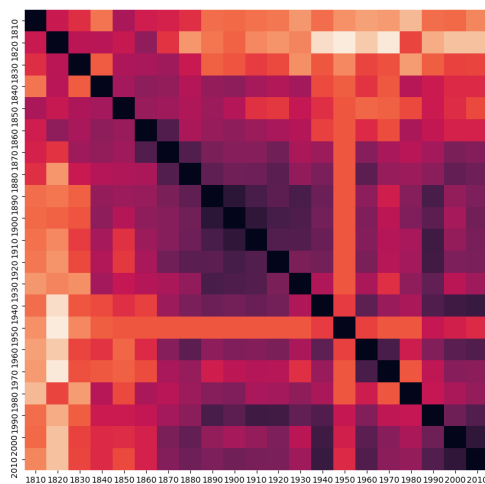
¹Инструмент доступен по ссылке: <https://github.com/text-processing/prose-rhythm-detector>



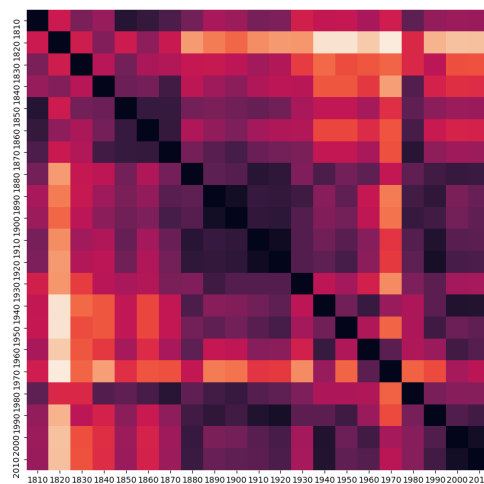
a)
Fig. 3. a) Word level, b) rhythm level, Chebyshev metric, Russian



b)
Рис. 3. а) Уровень слов, б) уровень ритма, метрика Чебышёва, русский язык



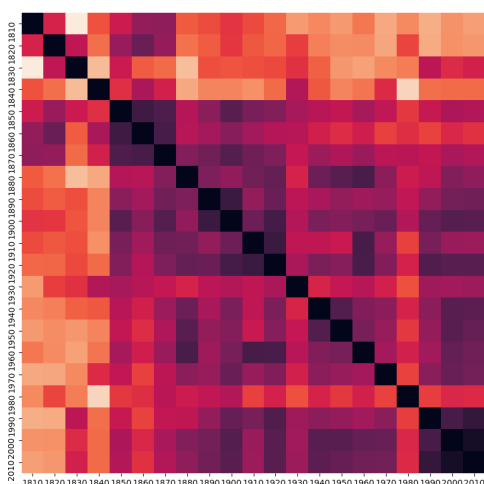
a)
Fig. 4. a) Symbol level, b) word level, Chebyshev metric, English



b)
Рис. 4. а) Уровень символов, б) уровень слов, метрика Чебышёва, английский язык

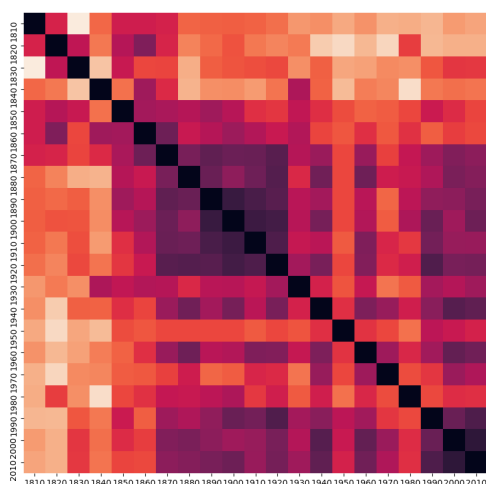
как и в случае с русской литературой, английская литература середины 19-го века выделяется на фоне всего остального столетия, пусть и не так сильно.

Наиболее показательными оказались результаты, полученные при расчёте по метрике Чебышёва. На картах, построенных по корпусу русского языка, отчётливо видно, как выделяются 1950-е годы (см. Рис. 2а)). Они отличаются от остальных периодов в литературе на всех уровнях, но особый вклад в это различие вносит уровень символов (см. Рис. 2б)) — все произведения рассматриваемой эпохи имеют сходство на уровне символов, в то время как 1950-е годы разительно отличаются от



a)

Fig. 5. a) Rhythm level, b) all levels, Chebyshev metric, English



b)

Рис. 5. а) Уровень ритма, б) все уровни вместе, метрика Чебышёва, английский язык

литературы всех периодов. На уровне слов (см. Рис. 3а)) можно выделить отличие текстов 21-го века от 19-го, а также стоит отметить различие 1950-х от второй половины 19-го века. Кроме того, на всех уровнях можно выделить схожесть 30-х и 40-х годов 19-го века. 1870-е года выделяются на фоне всей литературы 19-го века на всех уровнях, а уровень ритма (см. Рис. 3б)) придаёт им отличие от всей литературы за рассматриваемый период. В заключение, на уровнях символов и слов можно выделить близость второй половины 20-го века с началом 21-го века, однако уровень ритма слегка нивелирует это сходство.

Эксперименты на корпусе английских текстов показали следующие результаты. На уровне символов (см. Рис. 4а)) можно выделить близость 1890-х и 1900-х годов, а также 2000-х и 2010-х. Также на этом уровне особенно отличаются 1950-е годы по сравнению со всеми другими десятилетиями. На уровне слов (см. Рис. 4б)) уже более ярко выделяется период с 1890-х по 1920-е годы, а также 21-й век. Уровень ритма (см. Рис. 5а)) способствует выделению периода с 1990-х по 2010-е годы, а также периода с 1850-х по 1970-е годы и с 1890-х по 1900-е годы. Таким образом, на карте, отражающей все характеристики сразу (см. Рис. 5б)), мы четко можем выделить интервал с 1870-х по 1920-е годы, а также 21-й век и 1990-е годы.

4.3. Тепловые карты диапазонов

Второй тип тепловых карт (Рис. 6) отображает нормализованные значения характеристик (разность между реальным и средним значением по всему корпусу, делённую на среднеквадратическое отклонение) для обоих языков.

Первые 14 столбцов обозначают ритмические характеристики: 9 конкретных средств, общее количество появлений средств, доля слов, повторяющихся только один раз (unique word), доли частей речи. Остальные 7 — это несколько характеристик низкого уровня: количество букв, всех символов, слов и предложений, а также средние длины предложений в символах (avg by ch) и словах (avg by word) и средние длины слов (avg by word).

Для ритмических средств обоих языков повторяется тенденция, выявленная в предыдущем исследовании на меньших корпусах текстов: на протяжении веков общее число ритмических характеристик снижается. Это видно по оттенкам цветов на карте: для диакопы и многосоюзия

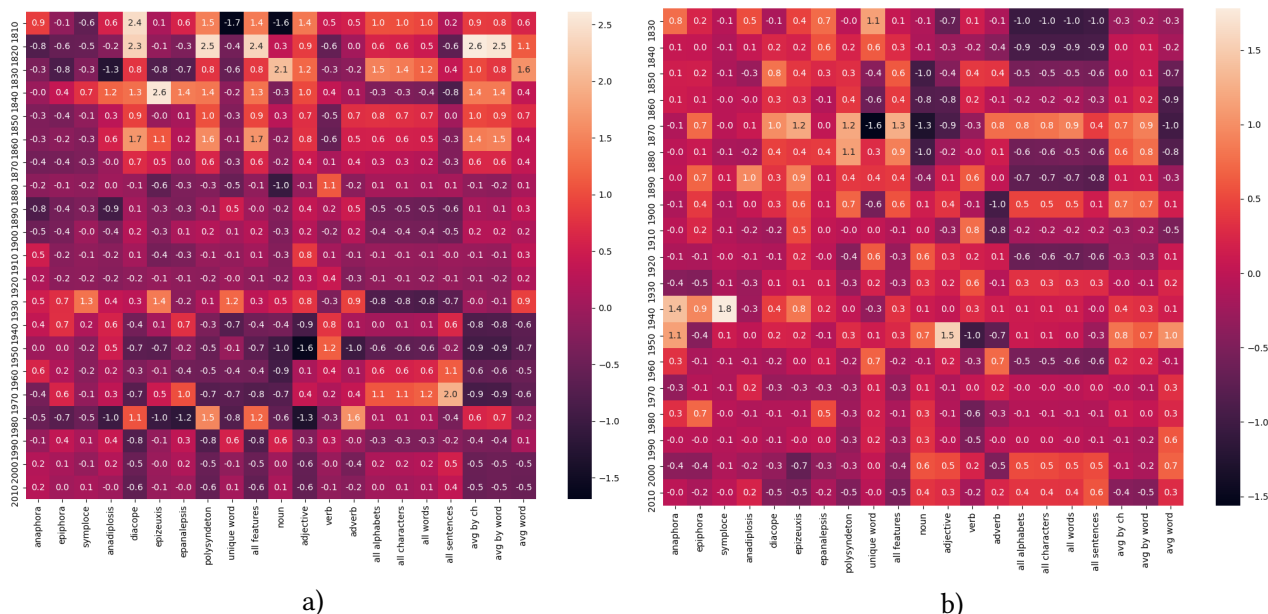


Fig. 6. Heatmap for a) English, b) Russian texts with normalized features

Рис. 6. Тепловая карта для а) английских, б) русских текстов с нормализованными значениями характеристик

(polysyndeton), как наиболее частых средств, и суммарного числа средств в 19 веке оттенки более светлые, а в конце 20-го — начале 21-го — более тёмные. Это значит, что в 19-м веке эти характеристики имеют значения выше среднего, а ближе к нашему времени — ниже среднего.

В британских текстах эта тенденция коррелирует с употреблением прилагательных: доля их появлений в ритмических средствах также снижается к 21-му веку. Доли остальных частей речи колеблются не так значительно. Для русских текстов закономерность другая: доли существительных и прилагательных в 20–21 веках возрастают.

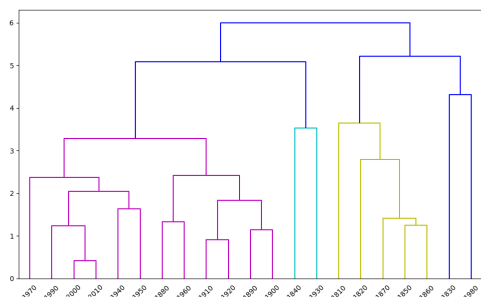
Что касается простых стилистических характеристик, то для британских текстов они показывают, как изменяется средний размер художественных произведений: постепенно уменьшается к середине 21-го века, затем снова возрастает. Для русских текстов тенденция обратная. Эти закономерности можно скорее отнести к особенностям формирования корпуса: были выбраны известные художественные произведения популярных авторов. При увеличении корпуса и добавлении более разнообразных произведений тенденции могут измениться.

Средние длины предложений и слов представляют собой характеристики, лучше отражающие стиль текстов, чем абсолютные количества элементов текста. Для британских текстов все они уменьшаются в течение практически всех десятилетий. Средняя длина предложений в русских текстах увеличивается к концу 19-го века — началу 20-го века, немного уменьшается в первой половине 20-го века, увеличивается к 1950-м годам, затем снова уменьшается. Средняя длина слова увеличивается на протяжении всех десятилетий.

Как и по тепловым картам близости, по картам диапазонов можно видеть десятилетия, выделяющиеся на фоне остальных. Причём на картах диапазонов дополнительно можно обнаружить, по каким стилистическим характеристикам десятилетия отличаются. Для британских текстов это 1930 и 1980 годы. Для русских — 1870 и 1940–1950 гг.

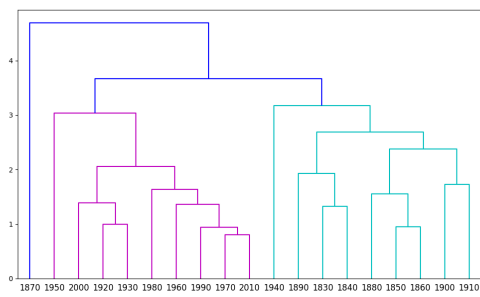
Таким образом, тепловые карты диапазонов позволяют как выявлять общие тенденции в изменении стиля произведений для языка в целом, так и обнаруживать отдельные десятилетия, выделяющиеся среди остальных.

4.4. Дендрограммы



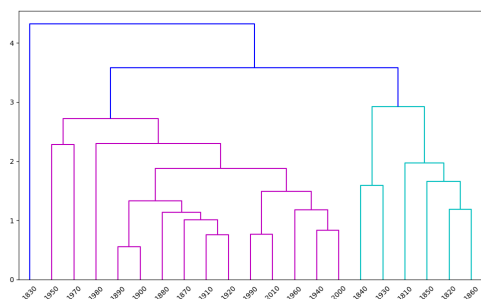
a)

Fig. 7. Dendrogram for a) English, b) Russian texts based on rhythm features, the Euclidean distance and complete-linkage method



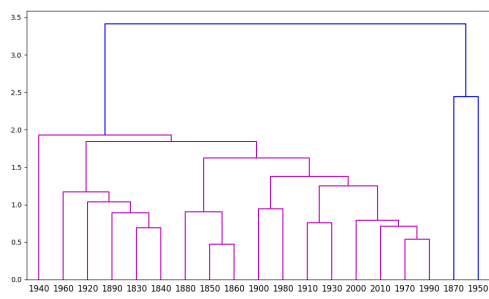
b)

Рис. 7. Дендрограмма для а) английских, б) русских текстов на основе ритмических средств, расстояния Евклида и метода полной связи



a)

Fig. 8. Dendrogram for a) English, b) Russian texts based on all features, the Chebyshev distance and complete-linkage method



b)

Рис. 8. Дендрограмма для а) английских, б) русских текстов на основе всех средств, расстояния Чебышёва и метода полной связи

Дендрограммы строились для обоих языков отдельно. Для каждого языка тексты кластеризовались иерархически как на основе отдельных типов стилометрических характеристик, так и на основе всех типов характеристик, чтобы сравнить разделение текстов только по ритму с разделением по всем стилометрическим характеристикам.

Среди функций расстояния между кластерами самые наглядные результаты показал метод полной связи, метод средней связи показал близкие к нему результаты. Метод одиночной связи выявил меньше кластеров, чем остальные.

Что касается метрик близости между элементами, то манхэттенское расстояние и расстояние Евклида показали близкие результаты, как и в случае с тепловыми картами. Коэффициент корреляции показал более хаотичное разбиение на кластеры, чем другие методы.

Для ритмических средств наиболее полезной оказалась метрика Евклида (см. Рис. 7).

Для британских текстов по ритму явно выделяются несколько небольших кластеров, содержащих соседние десятилетия: 1990–2010 гг., 1940–1950 гг., 1890–1920 гг., 1850–1870 гг. 1830 и 1980 десятилетия оказались самыми далёкими по ритму от остальных. 21 век наиболее похож на середину 20-го века: 1940–1950 гг. и 1970-й год.

Для русских текстов дендрограмма показывает меньшие расстояния по ритму между десятилетиями, чем для британских текстов. По ней видны два больших кластера, первый содержит большую часть десятилетий 20–21 веков, второй – 19-й век и начало 20-го: 1900–1910 гг. 21-й век не так явно выделяется по ритму, как в британских текстах. Самыми далёкими от остальных оказываются 1870-е и 1940-е десятилетия.

Для всех стилометрических средств наиболее наглядные результаты показала метрика Чебышёва (см. Рис. 8).

Для британских текстов по стилю некоторые пары соседних десятилетий снова оказываются близки друг к другу: 1890 и 1900, 1880 и 1870, 1910 и 1920 и т.д. В целом расстояния между десятилетиями оказываются меньше, в кластере 19-го века оказываются начало и середина этого столетия. 21-й век снова близок по стилю к середине 20-го века.

Для русских текстов по стилю 21-й век выделяется в отдельный кластер, но в этот кластер также попадают 1970-е годы. 19 век оказывается разбит на 2 более мелких кластера и более похож на 20-й век. 1940 и 1870 снова оказываются самыми далёкими от остальных, и к ним присоединяются 1950 гг.

Таким образом, дендрограммы показывают, что по ритмическим характеристикам века отличаются сильнее, чем по совокупности стилометрических характеристик. 19-й век и 1990–2010 годы могут выделяться в отдельные кластеры, 20-й век оказывается куда менее однородным как по ритму, так и по более простым стилометрическим характеристикам.

5. Обсуждение результатов

Автоматическое определение комплекса низкоуровневых и высокоуровневых стилометрических характеристик даёт возможность быстро проанализировать большое количество объёмных произведений и сделать качественные выводы об изменении стиля с течением времени. Этот подход позволяет эксперту за короткое время получить детализированную модель стиля художественного текста.

Эксперименты показали, что, хотя десятилетия можно успешно кластеризовать по близости друг к другу, каждое из них является уникальным по совокупности ритмических и простых стилометрических характеристик. Это значит, что на основе модели, построенной на данных характеристиках, тексты потенциально можно успешно классифицировать по векам и десятилетиям создания/публикации, а также вычислять год публикации текста.

Кроме того, анализ текстов средствами кластеризации на основе тепловых карт и дендрограмм позволяет выявлять тенденции изменения стиля в литературе в целом, а также сравнивать литературы на разных языках между собой. В частности, для русской и британской литератур детектируется уменьшение числа ритмических средств на единицу текста (в данном случае предложение). Для русской литературы обнаружено, что средние длины предложений изменяются волнообразно в течение рассматриваемого периода, в то время как средние длины слов увеличиваются. В британской литературе средние длины как слов, так и предложений существенно уменьшаются. Если сравнивать между собой века обеих литератур, то 20-й оказывается самым разнородным с точки зрения стиля, а 21-й и 19-й отличаются между собой, но десятилетия внутри них достаточно похожи.

Помимо поиска общих тенденций, тепловые карты и дендрограммы позволяют обнаружить конкретные периоды времени, которые значительно отличаются от других. Это может быть интерпретировано как то, что в данный период попали один или несколько текстов, которые особенно сильно отличаются по стилю от современников. Таким образом в крупном корпусе текстов можно выявлять произведения с уникальным стилем.

Если сравнить между собой значимость стилометрических характеристик разных уровней, то можно сделать вывод, что и низкоуровневые, и ритмические достаточно полезны и могут обнаруживать одни и те же крупные кластеры десятилетий. Однако ритмические характеристики более разнородны, поэтому являются лучшими индикаторами уникальности стиля.

Таким образом, автоматизированное моделирование текстов при помощи стилометрических характеристик различных уровней позволяет анализировать и успешно сравнивать между собой литературы разных языков и эпохи их развития.

Заключение

Авторы провели исследование стилометрических характеристик разных уровней: символов, слов и ритма на двух корпусах художественных произведений русской и английской литературы 19–21 веков. Стилометрические характеристики считались полностью автоматически при помощи инструмента ProseRhythmDetector². Он позволяет автоматически находить и статистически обрабатывать ритмические средства в комплексе с простыми стилометрическими характеристиками текста, что даёт возможность проводить анализ стиля прозы с разных сторон и исследовать большие корпуса текстов.

Анализ стилометрических характеристик позволил выявить как основные тенденции изменения стиля на протяжении 19–21 веков, так и обнаружить периоды времени, наиболее отличающиеся от других по ритму и стилю текстов. Кроме того, исследование показало значимость ритмических характеристик как маркеров особенностей стиля прозы.

Следующим этапом исследования стилометрических характеристик текста, в том числе и ритмических, может быть их использование для классификации текстов по веку или эпохе публикации, определения авторства, а также анализ и сравнение стилей литератур других языков.

References

- [1] E. Boychuk, I. Paramonov, N. Kozhemyakin, and N. Kasatkina, “Automated approach for rhythm analysis of French literary texts”, in *Proceedings of 15th Conference of Open Innovations Association FRUCT*, IEEE, 2014, pp. 15–23.
- [2] N. Golubeva-Monatkina, “On the Problem of Prose Rhythm”, *The Bulletin of the Russian Academy of Sciences: Studies in Literature and Language*, vol. 76, no. 2, pp. 16–27, 2017, In Russian.
- [3] T. Neal, K. Sundararajan, A. Fatima, Y. Yan, Y. Xiang, and D. Woodard, “Surveying stylometry techniques and applications”, *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, p. 86, 2018.
- [4] K. Lagutina, N. Lagutina, E. Boychuk, I. Vorontsova, E. Shliakhtina, O. Belyaeva, and I. Paramonov, “A Survey on Stylometric Text Features”, in *Proceedings of the 25th Conference of Open Innovations Association FRUCT*, IEEE, 2019, pp. 184–195.
- [5] Martynenko G. Ya., “Metody matematicheskoy lingvistiki v stilisticheskikh issledovaniyakh”, In Russian, Nestor-Istoriya, 2019.
- [6] A. Kumar, M. Lease, and J. Baldridge, “Supervised language modeling for temporal resolution of texts”, in *Proceedings of the 20th ACM international conference on Information and knowledge management*, 2011, pp. 2069–2072.
- [7] A. Jatowt and R. Campos, “Interactive system for reasoning about document age”, in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 2471–2474.
- [8] O. Popescu and C. Strapparava, “Semeval 2015, task 7: Diachronic text evaluation”, in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015, pp. 870–878.

²Инструмент опубликован по ссылке: <https://github.com/text-processing/prose-rhythm-detector>

- [9] A. Gopidi and A. Alam, “Computational Analysis of the Historical Changes in Poetry and Prose”, in *Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change*, 2019, pp. 14–22.
- [10] H. Lan and J. Huang, “Chinese-English Cross-Lingual Text Clustering Algorithm based on Latent Semantic Analysis”, *Proceedings of Science*, pp. 1–7, 2017.
- [11] A. Esuli, A. Moreo, and F. Sebastiani, “Funnelling: A New Ensemble Method for Heterogeneous Transfer Learning and Its Application to Cross-Lingual Text Classification”, *ACM Transactions on Information Systems (TOIS)*, vol. 37, no. 3, pp. 1–30, 2019.
- [12] K. Lagutina, A. Poletaev, N. Lagutina, E. Boychuk, and I. Paramonov, “Automatic Extraction of Rhythm Figures and Analysis of Their Dynamics in Prose of 19th-21st Centuries”, in *26th Conference of Open Innovations Association (FRUCT)*, IEEE, 2020, pp. 247–255.

Features of the Algorithmic Implementation of Difference Analogues of the Logistic Equation with Delay

S. D. Glyzin¹, S. A. Kashchenko¹, A. O. Tolbey¹

DOI: [10.18255/1818-1015-2020-3-344-355](https://doi.org/10.18255/1818-1015-2020-3-344-355)

¹Centre of Integrable Systems, P. G. Demidov Yaroslavl State University, Sovetskaya str., 14, Yaroslavl, 150003, Russia.

MSC2020: 39A11, 34K18

Research article

Full text in Russian

Received August 11, 2020

After revision September 7, 2020

Accepted September 9, 2020

The logistic equation with delay or Hutchinson's equation is one of the fundamental equations of population dynamics and is widely used in problems of mathematical ecology. We consider a family of mappings built for this equation based on central separated differences. Such difference schemes are usually used in the numerical simulation of this problem. The presented mappings themselves can serve as models of population dynamics; therefore, their study is of considerable interest. We compare the properties of the trajectories of these mappings and the original equation with delay. It is shown that the behavior of the solutions of the mappings constructed on the basis of the central separated differences does not preserve, even with a sufficiently small value of the time step, the basic dynamic properties of the logistic equation with delay. In particular, this map does not have a stable invariant curve bifurcating under the oscillatory loss of stability of a nonzero equilibrium state. This curve corresponds in such mappings to the stable limit cycle of the original continuous equation. Thus, it is shown that such a difference scheme cannot be used for numerical modeling of the logistic equation with delay.

Keywords: logistic equation with delay; mapping; bifurcation.

INFORMATION ABOUT THE AUTHORS

Sergey D. Glyzin	orcid.org/0000-0002-6403-4061 . E-mail: glyzin@uniyar.ac.ru
correspondence author	Head of the Department of Computer Networks, professor.
Sergey A. Kashchenko	orcid.org/0000-0002-4846-6040 . E-mail: kasch@uniyar.ac.ru
	First Vice-Rector, professor.
Anna O. Tolbey	orcid.org/0000-0001-5668-3929 . E-mail: bekva@yandex.ru
	Associate Professor, PhD.

Funding: The author were supported by the Russian Foundation for Basic Research (project no. 18-29-10043).

For citation: S. D. Glyzin, S. A. Kashchenko, and A. O. Tolbey, "Features of the Algorithmic Implementation of Difference Analogues of the Logistic Equation with Delay", *Modeling and analysis of information systems*, vol. 27, no. 3, pp. 344-355, 2020.

Особенности алгоритмической реализации разностных аналогов логистического уравнения с запаздыванием

С. Д. Глызин¹, С. А. Кашченко¹, А. О. Толбей¹

DOI: [10.18255/1818-1015-2020-3-344-355](https://doi.org/10.18255/1818-1015-2020-3-344-355)

¹Центр интегрируемых систем, Ярославский государственный университет им. П. Г. Демидова, 150003, г. Ярославль, ул. Советская, 14.

УДК 517.962.24

Научная статья

Полный текст на русском языке

Получена 11 августа 2020 г.

После доработки 7 сентября 2020 г.

Принята к публикации 9 сентября 2020 г.

Логистическое уравнение с запаздыванием или уравнение Хатчинсона представляет собой одно из фундаментальных уравнений популяционной динамики и находит широкое применение в задачах математической экологии. В работе рассматривается семейство отображений, построенное для этого уравнения на основе центральных разделенных разностей. Такие разностные схемы обычно используются при численном моделировании данной задачи. Представленные отображения сами по себе могут служить моделями динамики популяций, поэтому их изучение представляет значительный интерес. В работе сопоставляются свойства траекторий данных отображений и исходного уравнения с запаздыванием. Показано, что поведение решений отображений, построенных на основе центральных разделенных разностей, не сохраняет, даже при достаточно малой величине шага по времени, основных динамических свойств логистического уравнения с запаздыванием. В частности, у этого отображения при колебательной потере устойчивости ненулевого состояния равновесия не бифурцирует устойчивая инвариантная кривая. Эта кривая соответствует в таких отображениях устойчивому предельному циклу исходного непрерывного уравнения. Тем самым показано, что такая разностная схема не может быть использована для численного моделирования логистического уравнения с запаздыванием.

Ключевые слова: логистическое уравнение с запаздыванием; отображение; бифуркации.

ИНФОРМАЦИЯ ОБ АВТОРАХ

Сергей Дмитриевич Глызин автор для корреспонденции	orcid.org/0000-0002-6403-4061 . E-mail: glyzin@uniyar.ac.ru заведующий кафедрой компьютерных сетей, доктор физ.-мат. наук, профессор.
Сергей Александрович Кашченко	orcid.org/0000-0002-4846-6040 . E-mail: kasch@uniyar.ac.ru первый проректор, доктор физ.-мат. наук, профессор.
Анна Олеговна Толбей	orcid.org/0000-0001-5668-3929 . E-mail: bekva@yandex.ru доцент кафедры компьютерных сетей, канд. физ.-мат. наук.

Финансирование: Работа выполнена при финансовой поддержке РФФИ в рамках научного проекта № 18-29-10043.

Для цитирования: S. D. Glyzin, S. A. Kashchenko, and A. O. Tolbey, "Features of the Algorithmic Implementation of Difference Analogues of the Logistic Equation with Delay", *Modeling and analysis of information systems*, vol. 27, no. 3, pp. 344-355, 2020.

1. Постановка задачи

1.1. Простейшие свойства логистического уравнения с запаздыванием

Среди фундаментальных моделей популяционной динамики особое место занимает логистическое уравнение с запаздыванием

$$\frac{du}{dt} = r[1 - u(t-1)]u, \quad (1)$$

в котором неотрицательная функция $u(t)$ моделирует нормированную плотность численности популяции, а положительный параметр r характеризует скорость ее роста. В уравнении (1) за счет запаздывания в правой части простейшим способом моделируется возрастная структура популяции. Исследованию решений этого уравнения посвящено значительное количество литературы (см. [1–6]). Для изложения полученных нами результатов понадобятся простейшие свойства его решений, полученные в этих работах.

Через $C[-1, 0]$ ниже обозначается пространство непрерывных на отрезке $[-1, 0]$ функций со стандартной нормой. Это пространство примем в качестве фазового, т.е. пространства начальных условий уравнения (1). Для дальнейших построений нам потребуются два утверждения о решениях уравнения (1).

1. Для уравнения (1) выполняется теорема существования и единственности решений, т.е. для каждого значения t_0 и каждой начальной функции $\varphi(s) \in C[-1, 0]$, при всех $t > t_0$, существует и единственное решение $u(t, \varphi)$ уравнения (1), для которого $u(t_0 + s, \varphi) = \varphi(s)$.

2. Бифуркации Андронова – Хопфа. Положим в (1)

$$r = \pi/2 + \varepsilon,$$

где $0 < \varepsilon \ll 1$, тогда в достаточно малой и не зависящей от ε окрестности состояния равновесия $u \equiv 1$ существует двумерное устойчивое локальное интегральное многообразие (см., например, [7, 8]). На этом многообразии уравнение (1) можно записать в виде скалярного комплексного обыкновенного дифференциального уравнения

$$\frac{dg}{dt} = (\varepsilon\alpha_1 + O(\varepsilon^2))g + (d + O(\varepsilon))g|g|^2 + O(|g|^5), \quad (2)$$

где $g(t)$ — скалярная комплекснозначная функция и

$$\alpha_1 = \left(1 + \frac{\pi^2}{4}\right)^{-1} \left(\frac{\pi}{2} + i\right), \quad d = -\frac{1}{10}[3\pi - 2 + i(\pi + 6)] \left(1 + \frac{4}{\pi^2}\right)^{-1}. \quad (3)$$

Сформулируем бифуркационную теорему.

Теорема 1. Существует такое $\varepsilon_0 > 0$, что при всех $\varepsilon \in (0, \varepsilon_0]$ уравнение (1) имеет орбитально асимптотически устойчивый цикл $u_0(t, \varepsilon)$, для которого имеет место асимптотическое представление

$$u_0(t, \varepsilon) = 1 + \sqrt{\varepsilon \frac{20}{\pi(3\pi - 2)}} \cos\left((\pi/2 - 2\pi\varepsilon/(3\pi - 2))t\right) + O(\varepsilon). \quad (4)$$

1.2. Построение разностных аппроксимаций

Переход от уравнения (1) к разностному уравнению можно выполнить несколькими различными способами. В первую очередь остановимся на самом простом и очевидном способе, когда разностная модель получается в результате замены производной по времени на разделенную разность. Фиксируем произвольное натуральное k и будем считать, что шаг по времени равен $1/k$. Тогда простейшими способами выбора разделенной разности являются разность «вперед» $\frac{u(t + 1/k) - u(t)}{1/k}$, «центральная» разность $\frac{u(t + 1/k) - u(t - 1/k)}{2/k}$ и разность «назад» $\frac{u(t) - u(t - 1/k)}{1/k}$.

Случаи «вперед» и «назад», а также разностная схема, получающаяся из (1) на основе учета вольтерровской структуры правой части этого уравнения, подробно рассмотрены в [9–11].

В данной работе, в свою очередь, рассматривается случай «центральной» разности, для которого имеем

$$\frac{u(t + 1/k) - u(t - 1/k)}{2/k} = r[1 - u(t - 1)]u(t). \quad (5)$$

Сразу отметим, что точность приближения производной с помощью данной формулы выше, чем в случаях разностей «вперед» и «назад». Полагая $t = n/k$, $n \in \mathbb{Z}$ и обозначая $u_n = u(n/k)$, приходим к разностному уравнению порядка $k + 1$

$$u_{n+1} = u_{n-1} + \frac{2r}{k}(1 - u_{n-k})u_n, \quad n \geq 0. \quad (6)$$

Уравнение (6) будем называть дискретным вариантом логистического уравнения с запаздыванием. Этому уравнению, как и уравнению (1), может быть придан некоторый биологический смысл: величина $u_n > 0$ может интерпретироваться как численность очередного поколения моделируемой популяции, а величина k — как номер возрастной группы, начиная с которой особи популяции вовлекаются в процесс ее воспроизводства.

Одним из наиболее важных свойств уравнения Хатчинсона является появление у него при $r = \pi/2$ устойчивых периодических решений. Полученные разностные уравнения должны не только сохранять это свойство, но и при увеличении k все более и более приближаться к непрерывной модели. В связи с этим рассмотрим колебательную потерю устойчивости единичного состояния равновесия модели (6) и изучим режимы, ответвляющиеся от устойчивого состояния равновесия при критических значениях параметров.

Наша цель состоит в том, чтобы показать, что в отличие от рассмотренных ранее разностных схем, у разностного уравнения, построенного на основе центральной разности, при колебательной потере устойчивости ненулевого состояния равновесия не бифурцирует устойчивая инвариантная кривая. Эта кривая соответствует в таких отображениях устойчивому предельному циклу исходного непрерывного уравнения (1). Ее отсутствие позволит показать, что разностная схема (6) не может быть использована для численного моделирования логистического уравнения с запаздыванием на больших промежутках времени.

2. Локальные свойства дискретной модели логистического уравнения

2.1. Устойчивость единичной неподвижной точки отображения. Свойства характеристического многочлена

Рассмотрим задачу устойчивости единичного состояния равновесия предложенной модели. Для этого в уравнении (6) сделаем сдвиг на состояние равновесия $u_n = 1 + v_n$ и линеаризуем полученные уравнения в нуле. В результате за устойчивость этого состояния будет отвечать следующее линейное разностное уравнение:

$$v_{n+1} = v_{n-1} - \frac{2r}{k} v_{n-k}. \quad (7)$$

Выпишем для полученного линеаризованного уравнения соответствующий ему многочлен устойчивости. В данном случае характеристический многочлен имеет вид

$$P(\lambda) \stackrel{\text{def}}{=} \lambda^{k+1} - \lambda^{k-1} + \frac{2r}{k}. \quad (8)$$

Характеристический многочлен (8) для задачи (6) имеет свойства, существенно отличающиеся при четном и нечетном значениях k . Выполнены следующие два простых утверждения.

Лемма 1. Пусть k — четное, тогда при всех значениях $r > 0$ многочлен (8) имеет как минимум один вещественный отрицательный корень по модулю больший единицы.

Доказательство этого факта следует из нечетности степени многочлена (8) и следующих его свойств:

$$\lambda^{k+1} - \lambda^{k-1} + \frac{2r}{k} \Big|_{\lambda=-1} = \frac{2r}{k} > 0,$$

в то же время

$$\lambda^{k+1} - \lambda^{k-1} + \frac{2r}{k} \rightarrow -\infty, \quad \text{при } \lambda \rightarrow -\infty.$$

Лемма 2. Пусть $k = 2m + 1$ ($m = 1, 2, \dots$), тогда

1. При выполнении неравенства

$$0 < r < r_*, \quad \text{где } r_* \stackrel{\text{def}}{=} (2m+1) \sin\left(\frac{\pi}{2(2m+1)}\right), \quad (9)$$

все корни многочлена (8) лежат в круге $S = \{\lambda \in \mathbb{C} : |\lambda| < 1\}$ комплексной плоскости, а при $r = r_*$ уравнение $P(\lambda) = 0$ имеет две простые пары корней

$$\lambda_{1,2} = \exp(\pm i\omega_1), \quad \text{где } \omega_1 = \frac{\pi}{2(2m+1)}, \quad \text{и } \lambda_{3,4} = \exp(\pm i\omega_2), \quad \text{где } \omega_2 = \pi - \omega_1, \quad (10)$$

а все остальные его корни по-прежнему находятся в S .

2. Корни (10) удовлетворяют требованиям отсутствия главных резонансов.

3. Выполняется неравенство

$$\frac{d}{d\varepsilon} |\lambda(\varepsilon)| \Big|_{\varepsilon=0} > 0, \quad (11)$$

где $\lambda = \lambda(\varepsilon)$ — корень многочлена (8), существующий при $r = r_* + \varepsilon$, $0 \leq \varepsilon \ll 1$ и обращающийся в $\exp(i\omega_j)$ $j = 1, 2$ при $\varepsilon = 0$.

4. При условии выполнения неравенства

$$0 < r < r_{**}, \quad \text{где} \quad r_{**} \stackrel{\text{def}}{=} \frac{2m+1}{2(m+1)} \left(\frac{m}{m+1} \right)^m, \quad (12)$$

многочлен $P(\lambda)$ имеет вещественные корни.

Доказательство. Многочлен (8) при $r = 0$ имеет корень $\lambda = 0$ кратности $k - 1$ и два простых корня $\lambda = 1$ и $\lambda = -1$. Обозначим через $\lambda_+(r)$ и $\lambda_-(r)$ корни этого многочлена, обращающиеся при $r = 0$ в единицу и минус единицу соответственно. Будем считать, что $0 < r \ll 1$. Нетрудно видеть, что $\lambda'_+(0) = -1/k < 0$ и $\lambda'_-(0) = 1/k > 0$. А это значит, что при малых r все корни уравнения $P(\lambda) = 0$ заведомо лежат в S .

Простой анализ функции (8) показывает, что при $k = 2m + 1$ она четная и имеет при $0 < r < r_{**}$ четыре вещественных корня. Если же $r > r_{**}$, то вещественных корней нет. Для определения критического значения $r = r_{**}$ необходимо, чтобы наряду с равенством $P(\lambda) = 0$, выполнялось равенство $P'(\lambda) = 0$. Данные два равенства позволяют найти $r_{**} = \frac{2m+1}{2(m+1)} \left(\frac{m}{m+1} \right)^m$ и соответствующую ему пару корней кратности два $\lambda = \pm \sqrt{\frac{m}{m+1}}$.

Для того чтобы определить характер поведения комплексных корней многочлена (8) при увеличении параметра r , подставим в уравнение $P(\lambda) = 0$ соотношение $\lambda = \exp(i\omega)$, $\omega > 0$, умножим результат на $\exp[-i(2m+1)\omega]$ и в получившемся выражении разделим вещественную и мнимую части. В результате приходим к системе уравнений

$$\begin{aligned} \cos(2m+1)\omega &= 0, \\ \sin(\omega) - \frac{r}{2m+1} \sin(2m+1)\omega &= 0. \end{aligned} \quad (13)$$

Нетрудно заметить, что допустимые частоты ω , определяющиеся из (13), задаются формулами

$$\omega_n = \frac{\pi}{2(2m+1)} + \frac{2n\pi}{2m+1}, \quad n = 0, 1, \dots, \quad (14)$$

Полученные равенства позволяют определить наименьшее значение $r_* > 0$ параметра r , при котором две пары комплексных корней

$$\lambda_{1,2} = \exp(\pm i\omega_1), \quad \text{где} \quad \omega_1 = \frac{\pi}{2(2m+1)}, \quad \text{и} \quad \lambda_{3,4} = \exp(\pm i\omega_2), \quad \text{где} \quad \omega_2 = \pi - \omega_1, \quad (15)$$

многочлена (8) первый раз выходят на единичную окружность. Это значение имеет следующий вид:

$$r_* = (2m+1) \sin\left(\frac{\pi}{2(2m+1)}\right). \quad (16)$$

Свойство нерезонансности полученных корней, как и неравенство (11), очевидно, следует из этих же формул. Лемма доказана.

В рамках леммы 2 при стремлении m к бесконечности, в пределе получаются равенства

$$\lim_{m \rightarrow \infty} r_*(m) = \lim_{m \rightarrow \infty} (2m+1) \sin\left(\frac{\pi}{2(2m+1)}\right) = \frac{\pi}{2}, \quad \lim_{m \rightarrow \infty} r_{**}(m) = \lim_{m \rightarrow \infty} \frac{2m+1}{2(m+1)} \left(\frac{m}{m+1} \right)^m = e^{-1}, \quad (17)$$

показывающие близость свойств многочлена (8) и квазимногочлена уравнения Хатчинсона (1) (см. [4, 5]).

2.2. Построение нормальной формы задачи

Доказанная в лемме 2 нерезонансность корней многочлена (8) позволяет при

$$r = r_* + \varepsilon, \quad 0 < \varepsilon \ll 1 \quad (18)$$

воспользоваться аналогом известной бифуркационной теоремы Андронова-Хопфа для отображений (см. [12], [13]). У данной бифуркации существуют и другие общепринятые названия: бифуркация рождения тора, бифуркация Неймарка-Сакера и даже бифуркация Андронова-Хопфа-Неймарка-Сакера. С бифуркацией Андронова-Хопфа для потоков эту бифуркацию роднит то, что в случае двумерных отображений она связана с рождением замкнутой инвариантной кривой из неподвижной точки (дискретный аналог предельного цикла). Следует отметить, что это соответствие довольно условное, поскольку в такой бифуркации очень важную роль играют резонансы. В случаях, когда эти резонансы слабые или величина частоты иррациональна, то аналогия достаточно полная: здесь, как и в случае потоков, из неподвижной точки рождается единственная замкнутая инвариантная кривая. В случае же так называемых главных резонансов, бифуркации совсем другие, см., например, [13–15]. Бифуркация Андронова-Хопфа для отображений была впервые исследована в работе Ю.И. Неймарка [16], который, однако, не рассмотрел случай сильных резонансов. Впоследствии этот пробел был устранен Р. Сакером [17, 18]. Способы исследования такой бифуркации хорошо известны и изложены в целом ряде монографий и учебных пособий.

Ниже будем использовать несколько иной способ нахождения асимптотики бифурцирующих из состояния равновесия решений на основе построения нормальной формы с непрерывным медленным временем (см. [9–11]).

Для того, чтобы выяснить характер потери устойчивости единичного состояния равновесия уравнения (6) при условии (18), рассмотрим $(k+1)$ -мерное отображение

$$\begin{aligned} y &= (y_0, y_1, \dots, y_k) \rightarrow (\bar{y}_0, \bar{y}_1, \dots, \bar{y}_k), \\ \bar{y}_s &= y_{s+1}, \quad s = 0, \dots, k-1, \quad \bar{y}_k = y_{k-1} + \frac{2(r_* + \varepsilon)}{k} (1 - y_0)y_k. \end{aligned} \quad (19)$$

Из линейного анализа и, в частности, из леммы 2 следует, что в некоторой достаточно малой окрестности состояния равновесия $y_0 = \dots = y_k = 1$ это отображение имеет экспоненциально устойчивое четырехмерное инвариантное многообразие. После перемещения этого состояния равновесия в начало координат и последующего перехода к подходящим криволинейным полярным координатам отображение (19) записывается в виде

$$\begin{aligned} \bar{\xi}_1 &= (1 + \varepsilon\alpha_1)\xi_1 + (a_{11}\xi_1^2 + a_{12}\xi_2^2)\xi_1 + O(\varepsilon^2|\xi| + \varepsilon|\xi|^2 + |\xi|^4), \\ \bar{\xi}_2 &= (1 + \varepsilon\alpha_2)\xi_2 + (a_{21}\xi_1^2 + a_{22}\xi_2^2)\xi_2 + O(\varepsilon^2|\xi| + \varepsilon|\xi|^2 + |\xi|^4), \\ \bar{\varphi}_1 &= \varphi_1 + \omega_1 + \varepsilon\beta_1 + b_{11}\xi_1^2 + b_{12}\xi_2^2 + O(\varepsilon^2 + \varepsilon|\xi| + |\xi|^3), \\ \bar{\varphi}_2 &= \varphi_2 + \omega_2 + \varepsilon\beta_2 + b_{21}\xi_1^2 + b_{22}\xi_2^2 + O(\varepsilon^2 + \varepsilon|\xi| + |\xi|^3), \end{aligned} \quad (20)$$

где $\alpha_j, \beta_j, a_{jk}, b_{jk}, j = 1, 2, k = 1, 2$ — некоторые вещественные постоянные, $|\xi|^2 = \xi_1^2 + \xi_2^2$.

После замены $\xi_j = \sqrt{\varepsilon}\rho_j, j = 1, 2$ отображение (20) приобретает вид

$$\begin{aligned} \bar{\rho}_1 &= \rho_1 + \varepsilon(\alpha_1 + a_{11}\rho_1^2 + a_{12}\rho_2^2)\rho_1 + O(\varepsilon^{3/2}), \\ \bar{\rho}_2 &= \rho_2 + \varepsilon(\alpha_2 + a_{21}\rho_1^2 + a_{22}\rho_2^2)\rho_2 + O(\varepsilon^{3/2}), \\ \bar{\varphi}_1 &= \varphi_1 + \omega_1 + \varepsilon(\beta_1 + b_{11}\rho_1^2 + b_{12}\rho_2^2) + O(\varepsilon^{3/2}), \\ \bar{\varphi}_2 &= \varphi_2 + \omega_2 + \varepsilon(\beta_2 + b_{21}\rho_1^2 + b_{22}\rho_2^2) + O(\varepsilon^{3/2}). \end{aligned} \quad (21)$$

Наряду с отображением (21), рассмотрим систему обыкновенных дифференциальных уравнений

$$\begin{aligned}\frac{d\rho_1}{dt} &= \varepsilon(\alpha_1 + a_{11}\rho_1^2 + a_{12}\rho_2^2)\rho_1, \\ \frac{d\rho_2}{dt} &= \varepsilon(\alpha_2 + a_{21}\rho_1^2 + a_{22}\rho_2^2)\rho_2, \\ \frac{d\varphi_1}{dt} &= \omega_1 + \varepsilon(\beta_1 + b_{11}\rho_1^2 + b_{12}\rho_2^2), \\ \frac{d\varphi_2}{dt} &= \omega_2 + \varepsilon(\beta_2 + b_{21}\rho_1^2 + b_{22}\rho_2^2),\end{aligned}\tag{22}$$

тогда отображение сдвига по ее траекториям за время $t = 1$ с точностью до величин порядка $\varepsilon^{3/2}$ совпадает с (21), а сама система (22) после замен $\varphi_j = \psi_j + \omega_j t$, $w_j = \rho_j \exp(i\psi_j)$, $j = 1, 2$, $\tau = \varepsilon t$ принимает вид

$$\begin{aligned}\frac{dw_1}{d\tau} &= \gamma_1 w_1 + (d_{11}|w_1|^2 + d_{12}|w_2|^2)w_1, \\ \frac{dw_2}{d\tau} &= \gamma_2 w_2 + (d_{21}|w_1|^2 + d_{22}|w_2|^2)w_2,\end{aligned}\tag{23}$$

где $\gamma_j = \alpha_j + i\beta_j$, $d_{jk} = a_{jk} + ib_{jk}$, $j = 1, 2$, $k = 1, 2$.

Для вычисления коэффициентов γ_j , d_{jk} рассмотрим вытекающее из (6) разностное уравнение с непрерывным временем (см. аналогичные вычисления в [9–11])

$$u(t+1) = u(t-1) + \frac{2(r_* + \varepsilon)}{k} [1 - u(t-k)]u(t), \quad t \in \mathbb{R},\tag{24}$$

формальное решение которого будем искать в виде ряда

$$u = 1 + \sqrt{\varepsilon} h_0(t, \tau) + \varepsilon h_1(t, \tau) + \varepsilon^{3/2} h_2(t, \tau) + \dots, \quad \tau = \varepsilon t.\tag{25}$$

Здесь

$$h_0 = w_1(\tau) \exp(i\omega_1 t) + \overline{w_1}(\tau) \exp(-i\omega_1 t) + w_2(\tau) \exp(i\omega_2 t) + \overline{w_2}(\tau) \exp(-i\omega_2 t),\tag{26}$$

$w_j(\tau)$, $j = 1, 2$ — медленно меняющиеся комплексные амплитуды, которые будут определены на третьем шаге алгоритма, а h_1 , h_2 — некоторые тригонометрические полиномы аргументов $\omega_1 t$ и $\omega_2 t$ с зависящими от медленного времени τ коэффициентами.

После подстановки соотношений (18), (25) в (24) и приравнивания коэффициентов при $\sqrt{\varepsilon}$, ε и $\varepsilon^{3/2}$ приходим к серии линейных неоднородных разностных уравнений вида

$$L(h_0) = 0, \quad L(h_k) = f_k(t, \tau), \quad k = 1, 2,\tag{27}$$

где $L(h) \stackrel{\text{def}}{=} h(t+1, \tau) - h(t-1, \tau) + (2r_*/k)h(t-k, \tau)$, а переменная τ рассматривается как параметр. В силу (26) и очевидных соотношений

$$L(\exp(\pm i\omega_j t)) = 0, \quad j = 1, 2,\tag{28}$$

первое уравнение из (27) представляет собой верное равенство. Во втором из этих уравнений неоднородность f_1 , имеющая вид

$$f_1 = -\frac{2r_*}{k} h_0(t, \tau)h_0(t-k, \tau),\tag{29}$$

является линейной комбинацией вторых гармоник $\exp(\pm 2i\omega_j t)$, $j = 1, 2$, $\exp(i(\pm\omega_1 \pm \omega_2)t)$.

Приравнивание коэффициентов при $\varepsilon^{3/2}$ дает

$$f_2 = -h'_0(t+1, \tau) + h'_0(t-1, \tau) + r_* h'_0(t-k, \tau) - \frac{2}{k} h_0(t-k, \tau) - \\ - \frac{2r_*}{k} (h_0(t, \tau)h_1(t-k, \tau) + h_1(t, \tau)h_0(t-k, \tau)).$$

Неоднородность f_2 может быть преобразована к виду

$$f_2 = f_{2,1}(\tau) \exp(i\omega_1 t) + f_{2,2}(\tau) \exp(i\omega_2 t) + \text{к.с.} + F_*(t, \tau), \quad (30)$$

где под к.с. подразумевается комплексно сопряженное выражение, а в выражении $F_*(t, \tau)$ собраны третьи гармоники по ω_1 , ω_2 , которые не являются резонансными. Необходимым и достаточным условием разрешимости уравнения для h_2 в классе тригонометрических полиномов является отсутствие в (30) первых гармоник, т. е. равенства

$$f_{2,1}(\tau) \equiv 0, \quad f_{2,2}(\tau) \equiv 0. \quad (31)$$

Для выполнения данных равенств будем использовать еще не определенные комплексные амплитуды $w_1(\tau)$, $w_2(\tau)$ (см. (26)). Выполняя соответствующие вычисления, убеждаемся, что из (31) для $w_1(\tau)$, $w_2(\tau)$ получается система дифференциальных уравнений (23), в которой

$$y_j = -\frac{\exp(i\omega_j k)}{P'(i\omega_j)}, \quad j = 1, 2, \quad (32)$$

$$d_{11} = \frac{4(1+i)\sin^2 \omega_1}{P(2i\omega_1)P'(i\omega_1)}, \quad d_{12} = \frac{8(1+i)\sin^2 \omega_1}{P(i(\omega_1 + \omega_2))P'(i\omega_1)}, \\ d_{21} = \frac{8(1+i)\sin^2 \omega_1}{P(i(\omega_1 + \omega_2))P'(i\omega_2)}, \quad d_{22} = \frac{4(1+i)\sin^2 \omega_1}{P(2i\omega_2)P'(i\omega_2)},$$

где $\omega_1 = \frac{\pi}{2(2m+1)}$, $\omega_2 = \pi - \omega_1$, $P(\lambda)$ определяется формулой (8), а $P'(\lambda)$ обозначено выражение

$$P'(\lambda) = (k+1)\lambda^k - (k-1)\lambda^{k-2}.$$

Сделаем одно замечание. Выше вместо разностного уравнения (6) использовалось соответствующее уравнение с непрерывным временем (24) и был приведен формализм построения асимптотического разложения для решения специального вида. С точки зрения теории нормальных форм ситуация является достаточно сложной. Дело в том, что характеристический многочлен $P(\lambda)$ для линеаризованного (на положительном состоянии равновесия) уравнения с непрерывным временем имеет при $r = r_*$ не две пары корней на единичной окружности $\lambda_{\pm} = \exp(\pm i\omega_0)$, а две счетные серии пар корней

$$\lambda_{nj\pm} = \exp(i2\pi n \pm i\omega_j) \quad n = 1, 2, \dots, j = 1, 2.$$

Поэтому критический случай в задаче об устойчивости состояния равновесия имеет бесконечную размерность. Подобный критический случай был исследован в [19]. На основе анализа из [19, 20] получаем, конечно, тот же результат для нахождения коэффициентов y_j и d_{jk} .

Полярная замена $w_j = z_j \exp(i\psi_j)$, $j = 1, 2$ в системе (23) позволяет разделить амплитудные и фазовые переменные и рассмотреть систему амплитудных переменных отдельно

$$\begin{aligned}\frac{dz_1}{dt} &= \gamma_{10}z_1 + (a_{11}z_1^2 + a_{12}z_2^2)z_1, \\ \frac{dz_2}{dt} &= \gamma_{20}z_2 + (a_{21}z_1^2 + a_{22}z_2^2)z_2,\end{aligned}\tag{33}$$

где

$$\begin{aligned}\gamma_{10} = \gamma_{20} = \operatorname{Re} \gamma_1 &= \frac{(2m+1) \cdot \sin \omega_1}{2(\cos^2 \omega_1 + (2m+1)^2 \sin^2 \omega_1)}, \\ a_{11} = \operatorname{Re} d_{11} &= \frac{(-\cos \omega_1 - (2m+1) \sin(2\omega_1) - (2m+1) \sin \omega_1 + 2 \cos^2 \omega_1) \cdot \sin \omega_1}{(1 + 4 \cos^2 \omega_1) \cdot (\cos^2 \omega_1 + (2m+1)^2 \sin^2 \omega_1)}, \\ a_{12} = \operatorname{Re} d_{12} &= \frac{-2(\cos \omega_1 + (2m+1) \sin \omega_1) \cdot \sin \omega_1}{\cos^2 \omega_1 + (2m+1)^2 \sin^2 \omega_1}, \\ a_{21} = \operatorname{Re} d_{21} &= \frac{-2(-\cos \omega_1 + (2m+1) \sin \omega_1) \cdot \sin \omega_1}{\cos^2 \omega_1 + (2m+1)^2 \sin^2 \omega_1}, \\ a_{22} = \operatorname{Re} d_{22} &= \frac{(\cos \omega_1 - (2m+1) \sin(2\omega_1) + (2m+1) \sin \omega_1 + 2 \cos^2 \omega_1) \cdot \sin \omega_1}{(1 + 4 \cos^2 \omega_1) \cdot (\cos^2 \omega_1 + (2m+1)^2 \sin^2 \omega_1)}.\end{aligned}$$

Непосредственный анализ полученных коэффициентов позволяет обосновать следующее утверждение.

Лемма 3. *Выполнены неравенства*

$$\gamma_{10} = \gamma_{20} > 0, \quad a_{11} < 0, \quad a_{22} > 0.\tag{34}$$

Неравенства (34) гарантируют недиссипативность системы (33), а в след за ней и системы (23).

Это означает, что при $r = r_* + \varepsilon$, $0 < \varepsilon \ll 1$ в окрестности порядка $\sqrt{\varepsilon}$ единичного состояния равновесия разностного уравнения (6) отсутствуют устойчивые решения (см. [12, 13]).

3. Некоторые выводы

Подведем некоторые итоги. Специфика построенных разностных уравнений состоит в том, что они обладают другой динамикой по сравнению с уравнением Хатчинсона. В частности, характеристический многочлен уравнения (6), линеаризованного на единичном решении, имеет при четном k и произвольном положительном r отрицательный корень, по модулю больший единицы. Это влечет неустойчивость единичного решения при любом $r > 0$ и, более того, к тому, что компоненты u_n решения уравнения (6) с положительными начальными условиями становятся отрицательными. Это обстоятельство не позволяет при четном k привлекать модель (6) к анализу динамики уравнения с запаздыванием (1).

При нечетном k в уравнении (6) существует критическое значение параметра $r = r_*$ такое, что при $0 < r < r_*$ единичное состояние равновесия модели устойчиво, однако его способ потери устойчивости существенно отличается от полученного ранее у других моделей. Это связано с наличием у характеристического многочлена (8) при критических значениях r не одной, а двух пар корней на единичной окружности комплексной плоскости. При построении нормальной формы уравнения (6) выяснилось, что для выбранной аппроксимации это уравнение перестаёт адекватно моделировать

динамику уравнения Хатчинсона. В частности, при значениях параметра r , соответствующих потере устойчивости единичного состояния равновесия, от него не ответвляется устойчивая инвариантная кривая, которой у уравнения Хатчинсона соответствует устойчивый цикл (устойчивые периодические колебаний). Это показывает, что при выборе разностной схемы, моделирующей динамику уравнения Хатчинсона, не всегда удастся получить разностное уравнение с нужными свойствами.

Численный эксперимент для различных k показал, что при $r > r_*$ в этой модели не появляется устойчивой инвариантной кривой и решение уходит из области положительных значений.

References

- [1] E. M. Wright, "A non-linear difference-differential equation", *Journal für die reine und angewandte Mathematik*, vol. 194, pp. 66–87, 1955.
- [2] S. Kakutani and L. Markus, "On the non-linear difference-differential equation $y'(t) = (a - by(t - \tau))y(t)$ ", *Contributions to the Theory of Nonlinear Oscillations*, vol. 4, pp. 1–18, 1958.
- [3] G. S. Jones, "The existence of periodic solutions of $f'(x) = -\alpha f(x-1)\{1+f(x)\}$ ", *Journal of Mathematical Analysis and Applications*, vol. 5, pp. 435–450, 1962.
- [4] S. Kashchenko, "Asymptotics of the Solutions of the Generalized Hutchinson Equation", *Automatic Control and Computer Sciences*, vol. 47, no. 7, pp. 470–494, 2013.
- [5] S. A. Kashchenko, "Periodic Solutions of Nonlinear Equations Generalizing Logistic Equations with Delay", *Math. Notes*, vol. 102, pp. 181–192, 2017.
- [6] S. Kashchenko and D. Loginov, "About Global Stable of Solutions of Logistic Equation with Delay", *Journal of Physics: Conference Series*, vol. 937, no. 1, p. 012 019, 2017.
- [7] J. K. Hale, *Theory of functional differential equations*. New York: I Springer Verlag, 1977.
- [8] P. Hartman, *Ordinary Differential Equations*. New York: Wiley, 1964.
- [9] S. D. Glyzin and S. A. Kashchenko, "Finite-Dimensional Mappings Describing the Dynamics of a Logistic Equation with Delay", *Doklady Mathematics*, vol. 100, no. 1, pp. 380–384, 2019.
- [10] S. Glyzin and S. Kashchenko, "A family of finite-dimensional maps induced by a logistic equation with a delay", *Mathematical modeling*, vol. 32, no. 3, pp. 19–46, 2020.
- [11] S. D. Glyzin, A. Y. Kolesov, and N. K. Rozov, "Finite-dimensional models of diffusion chaos", *Comput. Math. and Math. Phys.*, vol. 50, pp. 816–830, 2010.
- [12] J. E. Marsden and M. McCracken, *The Hopf bifurcation and its applications*. Appl. Math. Sci., 19. Springer-Verlag, 1976.
- [13] È. È. Shnoĭ, "On the stability of fixed points of two-dimensional mappings", *Differ. Equ.*, vol. 30, no. 7, pp. 1156–1167, 1994.
- [14] V. I. Arnold, *Ordinary Differential Equations*. Springer-Verlag, 1992.
- [15] Y. A. Kuznetsov, *Elements of Applied Bifurcation Theory*. Springer-Verlag, 1995.
- [16] Y. I. Neimark, "On some cases of periodic motions depending on parameters (In Russian.)", *Dokl. Akad. Nauk SSSR*, vol. 129, pp. 736–739, 1959.
- [17] R. J. Sacker, "On invariant surfaces and bifurcation of periodic solutions of ordinary differential equations", in *Report IMM-NYU 333*, New York University, 1964.
- [18] R. J. Sacker, "A new approach to perturbation theory of invariant surfaces", *Comm. Pure Appl. Math.*, vol. 18, pp. 717–732, 1965.

- [19] I. S. Kashchenko and S. A. Kashchenko, “Normal and quasinormal forms for systems of difference and differential-difference equations”, *Communications in Nonlinear Science and Numerical Simulation*, vol. 38, pp. 243–256, 2016.
- [20] I. S. Kashchenko and S. A. Kashchenko, “Analysis of Local Dynamics of Difference and Close to Them Differential-Difference Equations”, *Russ Math.*, vol. 62, pp. 24–34, 2018.

A Generalization of the Inference Rules for Join Dependencies in Databases

S. V. Zykin¹

DOI: [10.18255/1818-1015-2020-3-356-365](https://doi.org/10.18255/1818-1015-2020-3-356-365)

¹Sobolev Institute of Mathematics SB RAS, 4 Acad. Koptyug avenue, Novosibirsk, 630090, Russia.

MSC2020: 68P15

Research article

Full text in Russian

Received July 6, 2020

After revision September 7, 2020

Accepted September 9, 2020

In this paper a generalisation of the inference rules of the join dependencies that are used in the design of database schemas that meets the requirements of the fifth normal form is considered. In the previous works devoted to this problem, attempts to construct systems of the axioms of such dependencies based on inference rules are made. However, while the justification for the consistency (soundness) of the obtained axioms does not cause difficulties, the proof of completeness in general has not been satisfactorily resolved. First of all, this is due to the limitations of the inference rules themselves. This work focuses on two original axiom systems presented in the works of Sciore and Malvestuto. For the inclusion dependencies a system of rules that generalises existing systems and has fewer restrictions has been obtained. The paper presents a proof of the derivability of known systems of axioms from the presented inference rules. In addition, evidence of the consistency (soundness) of these rules is provided. The question of the completeness of the formal system based on the presented rules did not find a positive solution. In conclusion, the theoretical and practical significance of the inference rules for the join dependencies is noted.

Keywords: relational databases; join dependencies; inference rules.

INFORMATION ABOUT THE AUTHORS

Sergey V. Zykin | orcid.org/0000-0002-0576-2149. E-mail: szykin@mail.ru
correspondence author | Leading researcher, Doctor of Science.

Funding: The work was supported by the program of fundamental scientific researches of the SB RAS № I.5.1., project № 0314-2019-0020.

For citation: S. V. Zykin, "A Generalization of the Inference Rules for Join Dependencies in Databases", *Modeling and analysis of information systems*, vol. 27, no. 3, pp. 356-365, 2020.

Обобщение правил вывода для зависимостей соединения в базах данных

С. В. Зыкин¹

DOI: [10.18255/1818-1015-2020-3-356-365](https://doi.org/10.18255/1818-1015-2020-3-356-365)

¹Институт математики им. С. Л. Соболева СО РАН, пр. ак. Коптюга, 4, г. Новосибирск, 630090 Россия.

УДК 004.652.4

Научная статья

Полный текст на русском языке

Получена 6 июля 2020 г.

После доработки 7 сентября 2020 г.

Принята к публикации 9 сентября 2020 г.

В работе рассматривается обобщение правил вывода зависимостей соединения, которые используются при проектировании схемы базы данных, удовлетворяющей требованиям пятой нормальной формы. В предшествующих работах, посвященных данной проблематике, делаются попытки построить системы аксиом таких зависимостей, основанных на правилах вывода. Однако, если обоснование непротиворечивости (надежности) полученных аксиом не вызывает затруднений, то доказательство полноты в общем случае не получило удовлетворительного решения. Прежде всего, это связано с ограниченностью самих правил вывода. В данной работе акцентировано внимание на двух оригинальных системах аксиом, представленных в работах Sciore и Malvestuto. Для зависимостей включения получена система правил, которая обобщает существующие системы и при этом имеет меньше ограничений. В работе представлено доказательство выводимости известных систем аксиом из представленных правил вывода. Кроме того, приводится доказательство непротиворечивости (надежности) этих правил. Вопрос о полноте формальной системы, основанной на представленных правилах, не нашел положительного решения. В заключение отмечена теоретическая и практическая значимость правил вывода для зависимостей соединения.

Ключевые слова: реляционные базы данных; зависимости соединения; правила вывода.

ИНФОРМАЦИЯ ОБ АВТОРАХ

Сергей Владимирович Зыкин
автор для корреспонденции

orcid.org/0000-0002-0576-2149. E-mail: szykin@mail.ru
вед. науч. сотр., доктор техн. наук.

Финансирование: Работа выполнена при поддержке программы фундаментальных научных исследований СО РАН № I.5.1., проект № 0314-2019-0020.

Для цитирования: S. V. Zykin, "A Generalization of the Inference Rules for Join Dependencies in Databases", *Modeling and analysis of information systems*, vol. 27, no. 3, pp. 356-365, 2020.

Введение

Основой проектирования логической структуры базы данных (БД) являются зависимости данных в прикладной области. В настоящее время актуальными при проектировании являются функциональные зависимости, многозначные зависимости, зависимости соединения [1–3] и зависимости включения [3–7]. Для рассмотрения материала введем необходимые обозначения. Пусть задана конечная совокупность всех отношений реляционной БД: $(R_1, R_2, \dots, R_k)$ – результат декомпозиции при нормализации отношений, $[R_i]$ – совокупность атрибутов (схема отношения R_i), $R_i[V]$ – проекция отношения R_i на множество атрибутов V , $U = \{A_1, A_2, \dots, A_n\}$ – конечное множество всех атрибутов, на которых заданы отношения БД.

Данная работа посвящена исследованию свойств зависимостей соединения, необходимых при построении пятой нормальной формы. Для того, чтобы подчеркнуть практическую значимость указанной проблемы, рассмотрим пример, позаимствованный из аналитических материалов С. Д. Кузнецова. Пусть A_1 – сотрудник, A_2 – проект, A_3 – этап проекта. Если предположить наличие зависимости соединения $\bowtie (A_1A_2, A_2A_3)$, где $\bowtie$ – знак операции естественного соединения [1, 2], то в результате декомпозиции отношения $R[A_1, A_2, A_3]$ будет получено два отношения:

$$R_1 = R[A_1, A_2], R_2 = R[A_2, A_3].$$

Наличие зависимости $\bowtie (A_1A_2, A_2A_3)$ гарантирует выполнение равенства

$$R = R_1 \bowtie R_2$$

для любой реализации R . Соответствующее зависимости бизнес-правило можно сформулировать следующим образом: “Если сотрудник участвует в каком-либо проекте, то он автоматически участвует в выполнении всех этапов этого проекта”. Заметим, что зависимость $\bowtie (A_1A_2, A_2A_3)$ эквивалентна многозначной зависимости $A_2 \twoheadrightarrow A_1|A_3$, следовательно, отношения декомпозиции (R_1, R_2) удовлетворяют четвертой нормальной форме (4НФ).

Если предположить наличие зависимости соединения $\bowtie (A_1A_2, A_2A_3, A_1A_3)$, то в результате декомпозиции отношения $R[A_1, A_2, A_3]$ будет получено три отношения:

$$R_1 = R[A_1, A_2], R_2 = R[A_2, A_3], R_3 = R[A_1, A_3].$$

Наличие зависимости $\bowtie (A_1A_2, A_2A_3, A_1A_3)$ гарантирует выполнение равенства

$$R = R_1 \bowtie R_2 \bowtie R_3$$

для любой реализации R . Соответствующее зависимости бизнес-правило: “Сотрудник участвует в выполнении только некоторых этапов проекта”, о чем записываются соответствующие данные в отношение R_3 . В отношениях R_1 и R_2 также должна быть информация об участии сотрудника в проекте и наличии соответствующего этапа в проекте. Отношения декомпозиции (R_1, R_2, R_3) удовлетворяют пятой нормальной форме (5НФ).

Допустим, что нам надо оценить участие сотрудника в проекте. Для реляционной схемы БД, сформированной по зависимостям, дополнения выполняются без разрушения существующей схемы. Для зависимости $\bowtie (A_1A_2, A_2A_3)$ надо добавить соответствующий атрибут в отношение R_1 . В этом случае оцениваться будет участие сотрудника во всем проекте. Для зависимости $\bowtie (A_1A_2, A_2A_3, A_1A_3)$ соответствующий атрибут должен быть добавлен в отношение R_3 . В этом случае оцениваться будет участие сотрудника в каждом этапе проекта.

Распространенным является утверждение, что зависимости соединения редко встречаются на практике. Рассмотренный пример опровергает это утверждение. Аналогичные зависимости возникают везде, где требуется сопоставить объекты с ограниченным множеством событий в системе.

1. Связанные работы

Риссанен [8] впервые ввел понятие зависимости соединения и определил важное свойство данных: соединение множества отношений $R_1, R_2, \dots, R_k$ без потери информации для любой реализации отношения R :

$$R = R_1 \bowtie R_2 \bowtie \dots \bowtie R_k,$$

что соответствует зависимости соединения $\bowtie (V_1, V_2, \dots, V_k)$, где $V_i = [R_i]$.

В работе [9] предложена система правил для зависимостей соединения. Пусть $S = \{X_1, X_2, \dots, X_p\}$, $R = \{Y_1, Y_2, \dots, Y_q\}$, $\text{attr}(S) = X_1 \cup X_2 \cup \dots \cup X_p$ и $\text{attr}(R) = Y_1 \cup Y_2 \cup \dots \cup Y_q$, $\models$ – логическое следствие, тогда:

A0: $\emptyset \models \bowtie (X)$ для любого множества $X \subseteq U$.

A1: $\bowtie (S) \models \bowtie (S, Y)$, если $Y \subseteq \text{attr}(S)$.

A2: $\bowtie (S, Y, Z) \models \bowtie (S, YZ)$.

A3: $\bowtie (S, Y), \bowtie (R) \models \bowtie (S, R)$, если $\text{attr}(R) = Y$.

A4: $\bowtie (S, YA) \models \bowtie (S, Y)$, если $A \not\subseteq \text{attr}(S)$.

В [9] показано, что правила A0 – A4 непротиворечивы (надежность – soundness), однако не обладают свойством полноты (completeness). Кроме того, рассмотрено обобщение рассмотренных правил при расширении множества U атрибутами специального вида.

В статье [10] сделана попытка найти надежную и полную формальную систему для зависимостей соединения. Получена ограниченная формальная система, позволяющая выводить зависимости соединения с использованием только обобщенных зависимостей соединения. Также получена неограниченная формальная система, которая полна для расширенных зависимостей соединения, получена надежная и полная неограниченная система в стиле Gentzen для зависимостей соединений. При этом остается открытым вопрос существования надежной и полной ограниченной формальной системы для расширенных зависимостей соединения.

Метод разложения зависимостей соединения в реляционной базе данных с использованием понятия шарнира (hinge) рассматривается в работе [11]. Показано, что метод декомпозиции может быть использован для повышения эффективности проверки целостности. Под проверкой целостности для зависимостей соединения подразумевается проверка выполнения этой зависимости в отношении после обновления. В общем случае такая задача является NP -полной. В работе показано, что этой проверки целостности можно избежать за счет разложения отношения на набор меньших отношений. Рассмотрено понятие n -цикличности для зависимостей соединения. Показано, что для фиксированного значения n проверка целостности может быть выполнена за полиномиальное время.

В работе [12] предложена надежная и полная формальная система для полных (full) ациклических зависимостей соединения, которая получена добавлением трех правил к аксиомам многозначных зависимостей. При определении ацикличности используется модель гиперграфа [2]. Аксиомы в этой системе запишем в следующем виде:

B1. $\bowtie (X_1, X_2) \models \bowtie (X_1 \cup Y, X_2 \cup Y)$.

B2. $\bowtie (X_1, X_2), \bowtie (Y, X_1 \cap X_2) \models \bowtie (X_1, X_2 \cap Y)$.

B3. $\models \bowtie (U)$.

B4. $\bowtie (X_1, X_2, \dots, X_k) \models \bowtie (Y_1, Y_2)$, где $\{Y_1, Y_2\}$ – компонент (двухреберный гиперграф).

B5. $\bowtie (X, \{X_1, X_2, \dots, X_k\}), \bowtie (Y_1, Y_2) \models \bowtie (X \cap Y_1, X \cap Y_2, \{X_1, X_2, \dots, X_k\})$, где $Y_1 \cap Y_2 \subseteq X, X \cap X_i \subseteq Y_1$ или $X \cap X_i \subseteq Y_2, i = 1, k$.

Далее мы вернемся к обсуждению этих аксиом.

Связь между аксиоматизацией зависимостей в реляционных базах данных и цилиндрическими алгебрами рассмотрена в работе [13]. Рассмотрены зависимости соединения и соответствующий класс (представимых) цилиндрических полурешеток. Поскольку представимые цилиндрические полурешетки имеют бесконечную аксиоматизируемую квазиэквивалентную теорию, то не существует конечной аксиоматизации для n -мерных неограниченных зависимостей соединения. В работе получены контрпримеры к утверждению, что цилиндрические зависимости конечно аксиоматизируемы.

В статьях [14, 15] рассматриваются полные иерархические зависимости (FHD), которые соответствуют естественному соединению, по крайней мере, двух отношений БД. В статье [14] определяется полный набор правил вывода для FHD. Первый результат устанавливает конечную аксиоматизацию для FHD. Вторым основным результатом устанавливается конечная аксиоматизация для выводимости FHD. В статье [15] показано, как выводы полных иерархических зависимостей можно моделировать с помощью выводов многозначных зависимостей и наоборот. Это позволяет применять оба основных результата к комбинированному классу функциональных и многозначных зависимостей. Кроме того, установлена новая аксиоматизация для класса нетривиальных функциональных зависимостей.

В работе [16] получено обобщение для задачи выводимости атомов соединения и атомов включения, что дает полную аксиоматизацию для проблемы выводимости встроенных зависимостей соединения и зависимостей включения в специальном случае. В более общем смысле статья представляет способ поиска аксиоматизации для различных коллекций зависимостей в БД с использованием методов математической логики.

Исследование ациклических зависимостей продолжено в работе [17]. Представлен формальный контекст для зависимостей соединения, который обобщает подход для анализа многозначных зависимостей. Этот результат является основой для дальнейшего исследования ациклических зависимостей соединения с целью минимизации их множества.

В данной работе рассматриваются только классические [8] зависимости соединения, которые не дополнены какими-либо ограничениями или расширениями. С точки зрения автора статьи, только такой вид зависимостей является значимым для практики.

2. Правила вывода зависимостей соединения

Пусть R – отношение, на множестве атрибутов $U = \{A_1, A_2, \dots, A_n\}$.

Определение 1. Зависимость соединения $\bowtie (X_1, X_2, \dots, X_k)$ выполнена, если для любой реализации отношения R выполнено:

$$R[X] = R[X_1] \bowtie R[X_2] \bowtie \dots \bowtie R[X_k],$$

где $R[X_i]$ – проекция [1] отношения R на множество атрибутов X_i , $X = X_1 \cup X_2 \cup \dots \cup X_k$.

В [2] сформулирован очевидный признак наличия зависимости соединения.

Утверждение 1. Зависимость $\bowtie (X_1, X_2, \dots, X_k)$ выполнена в R , если для любой реализации R наличие кортежей $t_1, t_2, \dots, t_k$, не обязательно различных, гарантирует наличие кортежа $t \in R$: $t[X_i] = t_i[X_i]$, $i = \overline{1, k}$, для которых $t_i[X_i \cap X_j] = t_j[X_i \cap X_j]$ если $X_i \cap X_j \neq \emptyset$.

Анализ утверждения 1 показывает, что разделение зависимостей соединения на полные (full) и встроенные является искусственным. В отношении R все полные зависимости соединения на практике могут оказаться встроенными в процессе эксплуатации БД. Это не должно стать причиной перестроения декомпозиции отношений. Произвольная зависимость $\bowtie (X_1, X_2, \dots, X_k)$ является полной в проекции $R[X]$, где $X = X_1 \cup X_2 \cup \dots \cup X_k$, и в общем случае встроенной в отношение R . В данном случае уместно говорить об области определения зависимости.

Определение 2. Областью определения зависимости соединения $\bowtie (X_1, X_2, \dots, X_k)$ в отношении R будем называть множество $X = X_1 \cup X_2 \cup \dots \cup X_k$.

В работе [18] было введено понятие области определения функциональной зависимости с целью разделения значений “не определено (Null)” и “не существует”. Также была разработана формальная теория, которая является естественным обобщением классической теории и позволяет избавиться от значений “не существует” на этапе проектирования схемы БД. Областью определения зависимости в [18] является множество атрибутов, что согласуется с определением 2. Таким образом, рассматриваемые далее зависимости являются полными, но в собственной области определения. Причем, выводимые зависимости могут иметь большую или меньшую область определения, чем исходные зависимости.

Рассмотрение формальных теорий в БД позволяет ответить на главные проблемы проектирования БД:

- является ли проект БД, построенный по множеству зависимостей, оптимальным и правильным относительно некоторых критериев (наличие аномалий, минимальное покрытие . . .),
- сохраняет ли декомпозиция $(R_1, R_2, \dots, R_k)$ зависимости в качестве ограничений целостности на данные.

В основе решения указанных проблем лежат правила вывода зависимостей. Рассмотрим множество правил для зависимостей соединения.

P0) $\emptyset \models \bowtie (X), X \subseteq U$.

P1) $\bowtie (X_1, X_2, \dots, X_k) \models \bowtie (Y_1, Y_2, \dots, Y_m)$, если:

- p11) для любого X_i существует Y_j : $X_i \subseteq Y_j$;
- p12) для любого Y_j выполнено $Y_j \subseteq X_1 \cup X_2 \cup \dots \cup X_k$.

P2) $\bowtie (X_1, X_2, \dots, X_k) \models \bowtie (Y_1, Y_2, \dots, Y_k)$, если:

- p21) $X_i \cap (X_1 \cup X_2 \cup \dots \cup X_{i-1} \cup X_{i+1} \cup \dots \cup X_k) \subseteq Y_i$;
- p22) $Y_i \subseteq X_i$.

P3) $\bowtie (X_1, X_2, \dots, X_k) \models \bowtie (Y_1, Y_2, \dots, Y_k)$, если:

- p31) $Y_i = X_i \cup Z$.

P4) $\bowtie (X_1, X_2, \dots, X_k, V), \bowtie (Y_1, Y_2, \dots, Y_m) \models \bowtie (X_1, X_2, \dots, X_k, V \cap Y_1, V \cap Y_2, \dots, V \cap Y_m)$, если:

- p41) $Y_i \cap Y_j \subseteq V, i \neq j$;
- p42) $V \cap (X_1 \cup X_2 \cup \dots \cup X_k) \subseteq V \cap (Y_1 \cup Y_2 \cup \dots \cup Y_m)$.

Рассмотрим соответствие представленных в данной работе правил.

Утверждение 2. Правила A0–A4 являются следствием правил P0–P4.

Доказательство. Очевидно, что правила A0 и P0 эквивалентны. Правило A1 является следствием P1 на множестве атрибутов $X = X_1 \cup X_2 \cup \dots \cup X_k$. Действительно, условие p12 правила P1 совпадает с условием $Y \subseteq \text{attr}(S)$ правила A1, а условие p11 правила P1 выполнено по построению правила A1. Поскольку P1 соответствует более общему случаю, то A1 является следствием P1 на множестве атрибутов $X = X_1 \cup X_2 \cup \dots \cup X_k$. Не сложно проверить, что $X = Y_1 \cup Y_2 \cup \dots \cup Y_k$. Правило A2 очевидное следствие правила P1. Правило A3 является частным случаем правила P4. Действительно, если $V = Y_1 \cup Y_2 \cup \dots \cup Y_m$, то условия p41 и p42 будут всегда выполнены, а само правило P4 будет

совпадать с правилом А3. Правило А4 становится очевидным следствием правила Р2, если сделать отождествления $S = (X_1, X_2, \dots, X_{i-1}, X_{i+1}, \dots, X_k)$ и $YA = X_i$. $\square$

Утверждение 3. *Правила В1–В5 являются следствием правил Р0–Р4.*

Доказательство. Очевидно, что правило Р3 является обобщением правила В1. Правило В2 выводимо с использованием правил Р4 и Р1. Действительно, запишем левую часть правила В2 в следующем виде $\bowtie (X, V), \bowtie (Y, X \cap V)$, необходимо получить вывод правила $\bowtie (X, V \cap Y)$. Подставив левую часть в правило Р4 получим выводимость $\bowtie (X, V \cap Y, V \cap X)$, условия р41 и р42 выполнены. Затем применим к полученному выражению правило Р1, получим выводимость $\bowtie (X, V \cap Y)$, условия р11 и р12 выполнены, что и требовалось показать. Правило В3 эквивалентно правилу Р0. Правило В4 является частным случаем правила Р1, поскольку компонент [12] является двухреберным гиперграфом, полученным объединением ребер исходного гиперграфа. Правило В5 является частным случаем правила Р4. Действительно, условие $Y_1 \cap Y_2 \subseteq X$ гарантирует выполнение условия р41 при $X = V$, а условие $X \cap X_i \subseteq Y_1$ или $X \cap X_i \subseteq Y_2, i = \overline{1, k}$, гарантирует выполнение условия р42. Поскольку порядок аргументов зависимости соединения не имеет значения, то правило В5 совпадает с правилом Р4, при $m = 2$. $\square$

3. Анализ надежности правил вывода

Для предложенных в данной работе правил Р0–Р4 необходимо доказать надежность (непротиворечивость). Другими словами, полученные за счет правил Р0–Р4 зависимости будут удовлетворять условию логического следствия. Это очень важная задача с практической точки зрения, поскольку гарантирует сохранение целостности данных при проектировании БД.

Теорема 1. *Правила Р0–Р4 являются надежными.*

Доказательство. Надежность правила Р0 очевидна.

Для доказательства правила Р1 обозначим: $R_X = R[X_1] \bowtie R[X_2] \bowtie \dots \bowtie R[X_k]$ и $R_Y = R[Y_1] \bowtie R[Y_2] \bowtie \dots \bowtie R[Y_m]$, где $X = X_1 \cup X_2 \cup \dots \cup X_k$ и $Y = Y_1 \cup Y_2 \cup \dots \cup Y_m$. Из леммы 5.5 [1] следует, что $R[X] \subseteq R_X$ и $R[Y] \subseteq R_Y$. Необходимо показать, что если $R_X \subseteq R[X]$, тогда $R_Y \subseteq R[Y]$. Рассмотрим произвольный кортеж $t_Y \in R_Y$, надо показать, что $t_Y \in R[Y]$. Для выполнения условия $t_Y \in R_Y$ необходимо наличие, не обязательно различных, кортежей $t_1, t_2, \dots, t_m$ в отношении R (утверждение 1), для которых выполнено $t_p[Y_p] = t_Y[Y_p], p = \overline{1, m}$. Рассмотрим любую пару множеств атрибутов X_i и X_j . Из условия р11 следует, что существуют p и $q: X_i \subseteq Y_p$ и $X_j \subseteq Y_q$. Если $X_i \cap X_j \neq \emptyset$, то $t_p[X_i \cap X_j] = t_q[X_i \cap X_j]$, поскольку они образованы из одного кортежа t_Y . Обозначим $t_{pi} = t_p[X_i]$. Из условия р11 следует, что кортеж t_{pi} существует для любого $i: 1 \leq i \leq k$. Из условия р12 следует, что не существует атрибута $A_l \in Y_p$, значение $t_Y[A_l]$ которого не принадлежит какому-либо кортежу t_{pi} . Естественное соединение кортежей t_{pi} совпадает с кортежем t_Y . Следовательно, $t_Y \in R_X$, заметим, что из условий р11 и р12 следует равенство $X = Y$. По предположению $R_X = R[X]$, а значит $t_Y \in R[Y]$. Правило доказано.

Для доказательства правила Р2 используем аналогичные обозначения: $R_X = R[X_1] \bowtie R[X_2] \bowtie \dots \bowtie R[X_k]$ и $R_Y = R[Y_1] \bowtie R[Y_2] \bowtie \dots \bowtie R[Y_k]$, где $X = X_1 \cup X_2 \cup \dots \cup X_k$ и $Y = Y_1 \cup Y_2 \cup \dots \cup Y_k$. Из леммы 5.5 [1] следует, что $R[X] \subseteq R_X$ и $R[Y] \subseteq R_Y$. Необходимо показать, что если $R_X \subseteq R[X]$, тогда $R_Y \subseteq R[Y]$. Рассмотрим произвольный кортеж $t_Y \in R_Y$, достаточно показать, что $t_Y \in R[Y]$. Для выполнения условия $t_Y \in R_Y$ необходимо наличие, не обязательно различных, кортежей $t_i \in R$, для которых выполнено $t_i[Y_i] = t_Y[Y_i], i = \overline{1, k}$. Рассмотрим произвольную пару множеств X_i и X_j , где $i \neq j$. Выполнено равенство $X_i \cap X_j \subseteq X_i \cap (X_1 \cup X_2 \cup \dots \cup X_{i-1} \cup X_{i+1} \cup \dots \cup X_k)$, из условия р21 следует $X_i \cap X_j \subseteq Y_i$. С другой стороны, $X_i \cap X_j \subseteq X_i \cap (X_1 \cup X_2 \cup \dots \cup X_{j-1} \cup X_{j+1} \cup \dots \cup X_k)$. Из условия р21 следует $X_i \cap X_j \subseteq Y_j$. В итоге получаем $X_i \cap X_j \subseteq Y_i \cap Y_j$ и $t_i[X_i \cap X_j] = t_j[X_i \cap X_j]$. По свойству операции естественного соединения существует кортеж $t_X = t_1[X_1] \bowtie t_2[X_2] \bowtie \dots \bowtie t_k[X_k] \in R_X$. При пустом

множестве $X_i \cap X_j$ отсутствует препятствие соединению кортежей. Из условия p22 следует $t_Y = t_X[Y]$. Поскольку $R_X = R[X]$ и $Y \subseteq X$, то $t_Y \in R[Y]$, что и требовалось доказать.

При доказательстве P3 используем прежние обозначения для R_X , R_Y , X , и Y . Как и прежде, $R[X] \subseteq R_X$ и $R[Y] \subseteq R_Y$. Необходимо показать, что если $R_X \subseteq R[X]$, тогда $R_Y \subseteq R[Y]$. Рассмотрим произвольный кортеж $t_Y \in R_Y$, необходимо показать, что $t_Y \in R[Y]$. Выполнение условия $t_Y \in R_Y$ гарантирует наличие, не обязательно различных, кортежей $t_i \in R$, для которых выполнено $t_i[Y_i] = t_Y[Y_i]$, $i = \overline{1, k}$. Эти кортежи удовлетворяют следующим условиям: $t_i[Y_i \cap Y_j] = t_j[Y_i \cap Y_j]$, $i = \overline{1, k}$. Поскольку $X_i \cap X_j \subseteq Y_i \cap Y_j$ и $Z \subseteq Y_i \cap Y_j$, $i, j = \overline{1, k}$, то кортеж $t_X = t_1[X_1] \bowtie t_2[X_2] \bowtie \dots \bowtie t_k[X_k]$ принадлежит R_X . Из условия $R_X \subseteq R[X]$ следует, что $t_X \in R[X]$ и существует кортеж $t \in R$, для которого $t_X = t[X]$. По совокупности рассмотренных равенств имеем $t_Y = t[Y]$, то есть $t_Y \in R[Y]$. Следовательно, $R_Y \subseteq R[Y]$, что и требовалось доказать.

Докажем надежность правила P4. Пусть $W_j = V \cap Y_j$, $j = \overline{1, m}$, $W = W_1 \cup W_2 \cup \dots \cup W_m$. Покажем, что $\bowtie (Y_1, Y_2, \dots, Y_m) \models \bowtie (W_1, W_2, \dots, W_m)$ выводимо. Для этого используем правило вывода P2. Условие p41 гарантирует выполнение условия $Y_i \cap Y_j \subseteq V \cap Y_j$, $i \neq j$, что, в свою очередь, гарантирует выполнение условия p21: $Y_j \cap (Y_1 \cup Y_2 \cup \dots \cup Y_{j-1} \cup Y_{j+1} \cup \dots \cup Y_k) \subseteq W_j = V \cap Y_j$. Условие p22 $W_j \subseteq Y_j$ выполнено по построению. Далее покажем, что $\bowtie (X_1, X_2, \dots, X_k, V) \models \bowtie (X_1, X_2, \dots, X_k, W)$ выводимо. Для этого также воспользуемся правилом P2. Из условия p42 следует условие p21: $V \cap (X_1 \cup X_2 \cup \dots \cup X_k) \subseteq W_j = V \cap (Y_1 \cup Y_2 \cup \dots \cup Y_m)$. Условие p22 $W_j \subseteq V$ выполнено по построению. Поскольку надежность правила P2 уже доказана, то выводимые с его помощью зависимости будут выполнимы. Осталось показать, что выполнима будет зависимость в правой части правила P4. По предположению текущая реализация отношения R удовлетворяет зависимостям $\bowtie (X_1, X_2, \dots, X_k, V)$ и $\bowtie (Y_1, Y_2, \dots, Y_m)$. Следовательно, R удовлетворяет зависимостям $\bowtie (X_1, X_2, \dots, X_k, W)$ и $\bowtie (W_1, W_2, \dots, W_m)$. Это означает, что:

$$\begin{aligned} R[X_1 \cup X_2 \cup \dots \cup X_k \cup W] &= R[X_1] \bowtie \\ &\bowtie R[X_2] \bowtie \dots \bowtie R[X_k] \bowtie R[W] \end{aligned}$$

и

$$R[W] = R[W_1] \bowtie R[W_2] \bowtie \dots \bowtie R[W_m],$$

подставим W и $R[W]$ в предыдущую формулу:

$$\begin{aligned} R[X_1 \cup X_2 \cup \dots \cup X_k \cup W_1 \cup W_2 \cup \dots \cup W_m] &= R[X_1] \bowtie R[X_2] \bowtie \\ &\dots \bowtie R[X_k] \bowtie R[W_1] \bowtie R[W_2] \bowtie \dots \bowtie R[W_m], \end{aligned}$$

что соответствует выполнимости зависимости $\bowtie (X_1, X_2, \dots, X_k, V \cap Y_1, V \cap Y_2, \dots, V \cap Y_m)$. $\square$

Замечание. При доказательстве теоремы не потребовалось предположение об ацикличности гиперграфов, соответствующих зависимостям соединения.

Для превращения множества правил в систему аксиом необходимо доказать полноту этой системы. К сожалению, все усилия исследователей в этом направлении пока не увенчались успехом.

Заключение

Полученная в работе система правил для зависимостей соединения имеет теоретическое значение с точки зрения развития теории проектирования баз данных. Для анализа полноты системы аксиом традиционно используется процедура прогонки [2], которая не дала удовлетворительных результатов как в этой работе, так и во всех предыдущих работах. Это означает, что поиск новых правил и обобщение известных правил должны быть продолжены. Полученные правила имеют ограниченное практическое значение при проектировании схемы БД. Действительно, после выделения функционально определяемых атрибутов в отдельные отношения при декомпозиции отношения R , либо при формировании обобщенного ключа (суперключа) при синтезе отношений, всякий раз приходится иметь дело с отношением, в котором нет функциональных зависимостей. Однако это отношение не является интерпретируемым. Использование зависимостей соединения позволяет найти корректную декомпозицию и получить интерпретируемые отношения и гарантировать выполнение свойства соединения без потерь информации.

References

- [1] J. Ullman, *Principles of Database Systems*. Stanford University: Computer Science Press, 1980.
- [2] D. Maier, *The Theory of Relational Databases*. Rockville: Computer Science Press, 1983.
- [3] S. Abiteboul, R. Hull, and V. Vianu, *Foundations of Databases: The Logical Level*. Addison-Wesley, 1994.
- [4] M. Casanova, R. Fagin, and C. Papadimitriou, "Inclusion dependencies and their interaction with functional dependencies", *J. Comput. Syst. Sci.*, vol. 28, no. 1, pp. 29–59, 1984. DOI: [10.1016/0022-0000\(84\)90075-8](https://doi.org/10.1016/0022-0000(84)90075-8).
- [5] M. Levene and M. Vincent, "Justification for inclusion dependency normal form", *IEEE Trans. Knowl. Data Eng.*, vol. 12, no. 2, pp. 281–291, 2000. DOI: [10.1109/69.842267](https://doi.org/10.1109/69.842267).
- [6] H. Köhler and S. Link, "Inclusion dependencies reloaded", in *The 24th ACM International on Conference on Information and Knowledge Management (CIKM '15)*, 2015, pp. 1361–1370. DOI: [10.1145/2806416.2806539](https://doi.org/10.1145/2806416.2806539).
- [7] V. Zykin and S. Zykin, "Analysis of Typed Inclusion Dependencies with Null Values", *Automatic Control and Computer Sciences*, vol. 52, no. 7, pp. 638–646, 2018. DOI: [10.3103/S0146411618070258](https://doi.org/10.3103/S0146411618070258).
- [8] J. Rissanen, "Theory of relations for databases – a tutorial survey", in *Mathematical Foundations of Computer Science 1978*, J. Winkowski, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 1978, pp. 536–551.
- [9] E. Sciore, "A Complete Axiomatization of Full Join Dependencies", *J. ACM*, vol. 29, no. 2, pp. 373–393, 1982. DOI: [10.1145/322307.322313](https://doi.org/10.1145/322307.322313).
- [10] C. Beeri and M. Vardi, "Formal systems for join dependencies", *Theoretical Computer Science*, vol. 38, pp. 99–116, 1985. DOI: [10.1016/0304-3975\(85\)90212-9](https://doi.org/10.1016/0304-3975(85)90212-9).
- [11] M. Gyssens, "On the complexity of join dependencies", *ACM Transactions on Database Systems*, vol. 11, no. 1, pp. 81–108, 1986. DOI: [10.1145/5236.5237](https://doi.org/10.1145/5236.5237).
- [12] F. Malvestuto, "A Complete Axiomatization of Full Acyclic Join Dependencies", *Inf. Process. Lett.*, vol. 68, no. 3, pp. 133–139, 1998.
- [13] I. Düntsch and S. Mikuláš, "Cylindric structures and dependencies in relational databases", *Theoretical Computer Science*, vol. 269, pp. 451–468, 2001. DOI: [10.1016/S0304-3975\(01\)00016-0](https://doi.org/10.1016/S0304-3975(01)00016-0).

- [14] S. Hartmann, H. Köhler, and S. Link, “Full hierarchical dependencies in fixed and undetermined universes”, *Annals of Mathematics and Artificial Intelligence*, vol. 50, pp. 195–226, 2007. doi: [10.1007/s10472-007-9067-0](https://doi.org/10.1007/s10472-007-9067-0).
- [15] J. Biskup and S. Link, “Appropriate inferences of data dependencies in relational databases”, *Annals of Mathematics and Artificial Intelligence*, vol. 63, no. 3–4, pp. 213–255, 2011.
- [16] M. Hannula and J. Kontinen, “A finite axiomatization of conditional independence and inclusion dependencies”, *Information and Computation*, vol. 249, pp. 121–137, 2016. doi: [10.1016/j.ic.2016.04.001](https://doi.org/10.1016/j.ic.2016.04.001).
- [17] J. Baixeries, “A Formal Context for Acyclic Join Dependencies”, *Lecture Notes in Computer Science*, vol. 10352, pp. 563–572, 2017. doi: [10.1007/978-3-319-60438-1_55](https://doi.org/10.1007/978-3-319-60438-1_55).
- [18] S. Zykin, “Domains of functional dependences in databases”, *Trudy Inst. Mat. i Mekh. UrO RAN*, vol. 22, no. 3, pp. 117–129, 2016.

