

МЕТОДИКА ПРЕПОДАВАНИЯ | TEACHING METHODS

Научная статья | Original paper

УДК 004.421.2:004.4:378.147

Учебный тренажёр по теории алгоритмов

Л.Н. Чернышов¹, В.Н. Лукин²✉

¹ Финансовый университет при Правительстве Российской Федерации
Москва, Российская Федерация

² Московский государственный психолого-педагогический университет
Москва, Российская Федерация

✉ lukinvn@list.ru

Резюме

Контекст и актуальность. В рабочих программах дисциплин «Теория алгоритмов», «Методы компиляции», «Дискретная математика» и других, входящих в образовательные программы специальностей по информационным технологиям, важное место занимают понятия алгоритма, конечных автоматов и формальных языков. Освоение этих понятий требует от студентов способностей к абстрактному мышлению и зачастую вызывают существенные затруднения. **Цель.** Повысить качество усвоения учебного материала по дисциплинам, связанным с понятием алгоритма, используя специальные программные средства для визуализации и моделирования различных конечных автоматов. **Гипотеза.** Использование учебных тренажеров для демонстрации моделей алгоритмов позволяет преподавателю максимально понятно подать материал, а студентам лучше усвоить материал и получить навыки решения задач по теории алгоритмов. **Методы и материалы.** В исследовании рассмотрены существующие программные средства для построения графов, диаграмм автоматов, машин Тьюринга. Анализ выявил недостатки отдельных средств, а именно: ограниченность возможностей, неудобный интерфейс, необходимость соблюдения лицензирования. **Результаты.** Разработано web-приложение «FSM-Designer», которое играет роль учебного тренажера по теории алгоритмов. Приложение имеет простой интерфейс, возможности построения и моделирования конечных автоматов, автоматов с магазинной памятью и машин Тьюринга. Опыт использования тренажера показал, что студенты лучше стали справляться с задачами. Кроме того, выяснилось, что студенты испытывали затруднения с использованием искусственного интеллекта при решении задач,



и это легко выявлялось. **Выводы.** Применение учебного тренажера подтвердило важность использования специальных программных средств для усвоения материала по теории алгоритмов.

Ключевые слова: теория алгоритмов, теория формальных языков и автоматов, преподавание информатики, инструментальные средства, моделирование автоматов, машина Тьюринга, использование искусственного интеллекта

Для цитирования: Чернышов, Л.Н., Лукин, В.Н. (2025). Учебный тренажёр по теории алгоритмов. *Моделирование и анализ данных*, 15(3), 161—171. <https://doi.org/10.17759/mda.2025150310>

Training simulator for algorithm theory

L.N. Chernyshov¹, V.N. Lukin²✉

¹ Financial University under the Government of the Russian Federation

Moscow, Russian Federation

² Moscow State University of Psychology and Pedagogy, Moscow, Russian Federation

✉ lukinvn@list.ru

Abstract

Context and relevance. In the work programs of the disciplines «Theory of Algorithms», «Compilation Methods», «Discrete Mathematics» and others included in the educational programs of specialties in information technology, the concepts of algorithm, finite automata and formal languages occupy an important place. Mastering these concepts requires abstract thinking abilities from students and often cause significant difficulties. **Purpose.** Improve the quality of assimilation of educational material in disciplines related to the concept of an algorithm using special software tools for visualization and modeling of various finite state machines. **Hypothesis.** The use of training simulators to demonstrate algorithm models allows the teacher to present the material as clearly as possible, and it is better for students to master the material and gain skills in solving problems in algorithm theory. **Methods and materials.** The study examined existing software tools for constructing graphs, diagrams of automata, Turing machines. The analysis revealed the shortcomings of individual tools, namely: limited capabilities, inconvenient interface, the need to comply with licensing. **Results.** A web application «FSM-Desiner» has been developed, which plays the role of a training simulator in algorithm theory. The application has a simple interface, the ability to build and simulate finite state machines, store memory machines and Turing machines. The experience of using the simulator showed that students began to cope with tasks better. In addition, it turned out that students had difficulty using artificial intelligence when solving problems, and this was easily detected. **Conclusions.** The use of the training simulator confirmed the importance of using special software tools for mastering the material on the theory of algorithms.



Keywords: theory of algorithms, theory of formal languages and automata, teaching computer science, tools, modeling automata, Turing machine, using artificial intelligence

For citation: Chernyshov, L.N., Lukin, V.N. (2025). Training simulator for algorithm theory. *Modelling and Data Analysis*, 15(3), 161—171. (In Russ.). <https://doi.org/10.17759/mda.2025150310>

Изучение теории алгоритмов включается в образовательные программы вузов по специальностям 09.02.03 «Программирование в компьютерных системах», 09.03.01 «Программная инженерия и компьютерные науки», 02.03.03 «Математическое обеспечение и администрирование информационных систем». Её цель — формирование знаний об основных результатах теории алгоритмов; развитие логической и алгоритмической интуиции как в математике, так и в информатике; формирование и развитие у студентов понимания уровня строгости математической модели. Здесь ключевая математическая модель — машина Тьюринга (МТ). В учебниках и учебных пособиях теме о конечных автоматах и МТ уделяется значительное внимание (Поляков, Скорубский, 2012), (Лукин, Чернышов, 2021), (Чернецкая, 2022).

Практика показывает, что студенты нередко испытывают трудности в изучении дисциплин, связанных с теорией автоматов. Причина кроется в том, что для этого требуется высокий уровень абстракции, в то время как автомат обычно ассоциируется с чем-то материальным. Поэтому у преподавателя стоит задача подать материал максимально наглядно без потери строгости. Здесь большую пользу может принести программное средство для демонстрации как статического представления автомата, так и его работу, что способствуют лучшему усвоению материала.

Базовые знания и теория в полном объеме даются в известных монографиях (Ахо, Ульман, 1978) и (Хопкрофт, Мотвани, Ульман, 2002).

Рассмотрим представление машины Тьюринга. Она задаётся перечислением символов алфавита, множеством состояний и набором правил перехода, по которым она работает. Диаграмму состояний можно определить в виде графа, на дугах которого отмечаются действия машины: перемещение головки по ленте и запись символа на ленту. Подобные диаграммы строятся и для других моделей вычислений.

Существует множество различных инструментов для построения диаграмм состояний, графов и машин Тьюринга (Редактор графов, 2025), (Эмулятор машины Тьюринга, 2025), (Finite Automata Designer, 2025), (Инструмент для построения диаграммы состояния, 2025) и др. Однако не все они удобны для демонстрации: как правило, нет возможности анимации действий не только машин Тьюринга, но и существенно более простых моделей: автоматов с магазинной памятью (МП-автоматов) и конечных автоматов (КА). Кроме того, они обычно предназначены для решения ограниченного набора задач для конкретных моделей и имеют сложный интерфейс.

Наиболее известный и популярный инструмент для визуализации объектов теории формальных языков — программа JFLAP, разрабатываемая в университете



Дьюка с 1990 года (JFLAB, 2018). Она имеет развитый функционал, позволяет моделировать автоматы разных типов, она мультиплатформенная (написана на языке Java). Но для её установки требуется регистрация (проприетарная лицензия). Кроме того, она достаточно объемная: более 300 папок и 6000 файлов, что затрудняет ее изучение и модификацию.

Для проведения практических занятий на построение автоматов и грамматик есть онлайн-сервис (AutomataTutor, 2025), распространяемый под лицензией Apache 2.0. Он позволяет преподавателям составлять упражнения, а студентам отправлять выполненные задания, который автоматически проверяются.

В курсе теории алгоритмов тема машин Тьюринга основная, но лишь одна из многих. Общим для всех видов автоматов является граф состояний. Существует множество различных программ для построения графов и решения задач с их помощью. Наиболее подходящей для использования оказалась программа с открытым кодом, разработанная И. Уоллесом (Wallace Evan, 2010). Программа имеет простой интерфейс и позволяет строить размеченные графы, в которых можно выделить конечные состояния. Построенные графы статичны и сохраняются в локальной памяти с помощью средства localStorage.

Однако для учебных целей необходима не только возможность визуализации автоматов. Нужна и анимация их работы, и пригодность для решения задач. Важно показать как формальное определение автоматов, так и динамику работы в виде последовательности конфигураций. Для машин Тьюринга есть программа (Эмулятор машины Тьюринга, 2025), разработанная именно для обучения, но в ней нет диаграмм. Диаграммы МТ строятся в системе (The busy beaver challenge, 2025), но для этого необходимо написать программу определения автомата на некотором псевдокоде.

Для использования в учебном процессе желательно иметь удобный инструмент, допускающий расширения для разных моделей и разных задач. Основой должен служить построитель графов, на элементы которого налагаются ограничения, соответствующие выбранной модели. Этот инструмент должен предоставлять преподавателю среду для подготовки демонстрационных примеров (презентаций) к лекциям, разработки заданий для контрольных и лабораторных работ. Кроме того, нужно, чтобы студенты, в свою очередь, могли использовать его при изучении соответствующих тем, решать задачи и готовить отчеты к ним.

Одним из авторов (Чернышовым Л.Н.) построено как раз такое программное средство: построитель конечных автоматов FSMD (см. www.chernyshov.com/FSM). Его основная цель — графическое представление конечных автоматов. В свою очередь, графическая форма может быть легко преобразована в текстовую. Это позволяет встраивать изображение в документы разных форматов: SVG — в HTML-документы, LaTeX — в офисные документы, JSON — в JavaScript-программы. Кроме того, предусмотрено преобразование графов в специальные представления, которые удобно использовать в программах на разных языках (в частности, в Haskell).

Основные возможности программы следующие:

- построение диаграмм конечных автоматов, автоматов с магазинной памятью, машин Тьюринга;



- экспорт диаграмм в JSON, SVG, LaTeX;
- импорт диаграмм (как отдельной диаграммы, так и набора) в JSON-формате;
- генерация по диаграммам формальных определений КА, МР-автомата, МТ, регулярных грамматик с проверкой корректности диаграмм;
- задание команд МТ в сокращенном виде (можно не указывать запись текущего символа и движение головки);
- задание МТ и автоматическая генерация диаграмм по компактному стандартному описанию;
- возможность определения детерминированности автоматов;
- нахождение пустых и недостижимых состояний в автоматах;
- сохранение рабочей диаграммы и комментария к ней в локальной памяти;
- формирование наборов диаграмм и хранение их в локальном хранилище;
- демонстрация пошаговой работы автоматов с визуализацией рабочей ленты (магазинной памяти для МР-автомата) и конфигураций как в ручном режиме, так и в автоматическом с заданием интервала времени;
- демонстрация вывода цепочек в регулярной грамматике;
- генерация случайных диаграмм;
- хранение на сервере демонстрационных наборов автоматов;
- доступ к базе данных МТ определенного вида для их исследования.

Для использования в учебном процессе созданы демонстрационные наборы автоматов. Они служат вспомогательным материалом при чтении лекций и на практических занятиях. Примеры, на которых они построены, взяты из монографий (Ахо, Ульман, 1978), (Хопкрофт, Мотвани, Ульман, 2002) и учебных пособий (Поляков, Скорубский, 2012), (Лукин, Чернышов, 2021). Там они снабжены комментариями и подробно разбираются.

Важная особенность интерфейса программы состоит в том, что на экране одновременно показаны формальное определение автомата, диаграмма состояний и последовательность конфигураций (рис. 1). Зачастую студенты с трудом могут дать формальное математическое определение автомата и результата его работы. Отображение различных форм одной и той же абстракции поможет понять суть и запомнить определения.

Программный код FSMD, как и ядро построителя диаграмм, открытый. Это даёт возможность ставить задачи по расширению возможностей и развивать силами студентов.

В программе предусмотрены параметры для настройки отображаемой ленты: ее размеры, число клеток на экране, ограничение с одного из концов и начальное состояние. Часть примеров МТ соответствует правилам, принятым в (Хопкрофт, Мотвани, Ульман, 2002), а именно: пустые клетки заполнены символом «B», который включается в алфавит МТ. При этом отмечается только клетки, которые расположены непосредственно слева и справа от обрабатываемой цепочки. В другом режиме лента полностью заполняется нулями, и МТ может начинать работу, когда на ленте нет символов, отличных от нуля.

В случае двухсимвольного алфавита $\{0,1\}$ МТ можно задать не только диаграммой, но и описанием в виде строки. Стока составляется их последовательности подстрок, разделенных знаком «_». Первая подстрока определяется командами перехода из начального состояния A в зависимости от 0 и 1 в текущей позиции на ленте.



Например, если $f(A,0) = (B, 1, R)$ и $f(A,1) = (C, 0, L)$, то строкой будет «1RB0LC». Далее следуют подстроки, полученные для состояний В. С и т. д. Последнее состояние считается конечным. Если перехода по какому-либо символу нет, соответствующая подстрока заменяется на «---». Диаграмма по такой строке, которая задается в комментариях, строится автоматически.

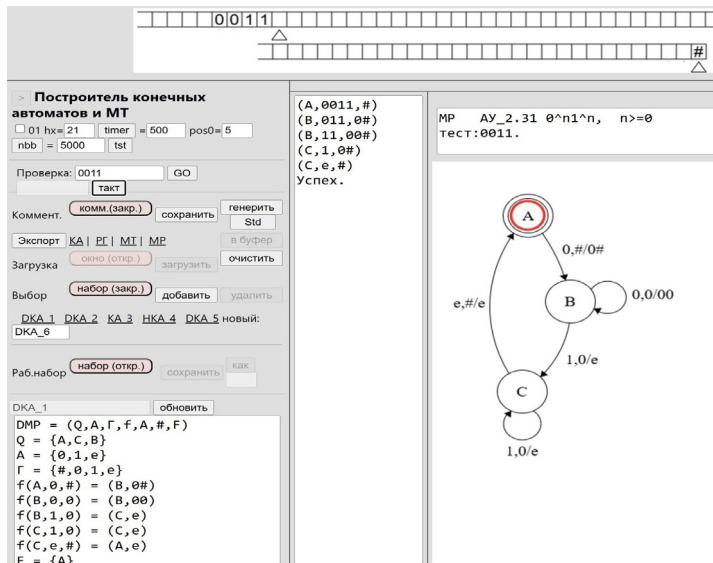


Рис. 1. Моделирование работы МП-автомата

Fig. 1. Simulation of pushdown automaton

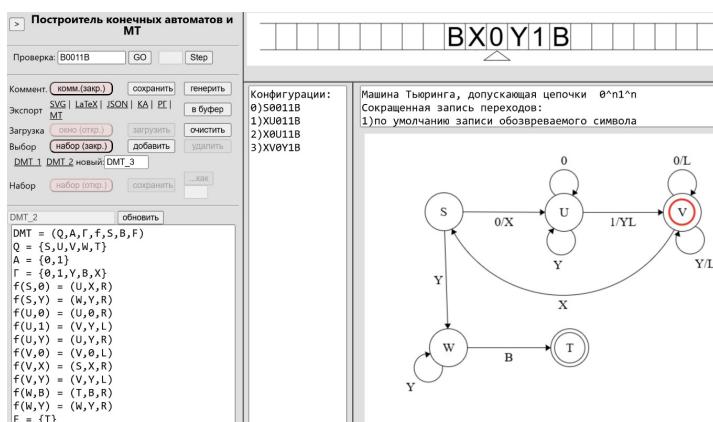


Рис. 2. Моделирование работы машины Тьюринга

Fig. 2. Simulation of Turing machine operation



Для МТ с начальной нулевой лентой естественным образом возникает задача, известная как проблема «усердного бобра» (The busy beaver challenge, 2025). Она состоит в том, чтобы среди всех МТ с n -состояниями и алфавитом {0,1} (их конечное число) найти машину BB(n), которая делает максимальное число шагов $S(n)$ до остановки и записывает на ленту максимальное число единиц $\Sigma(n)$.

Алан Тьюринг в 1936 году доказал, что не существует алгоритма, который для любой МТ мог бы определить, остановится ли она. Поэтому задача нахождения значений функций $S(n)$ и $\Sigma(n)$ для $n > 4$ оказывается непростой. Для $n=5$ задача была решена только в 2024 году. В программе FSMD приводится набор решений для n от 1 до 5, а для $n=6$ и $n=7$ построены наилучшие решения, обнаруженные в июне и мае 2025 года соответственно. На рисунке 3 приводятся диаграммы МТ для $n=5$ и $n=6$.

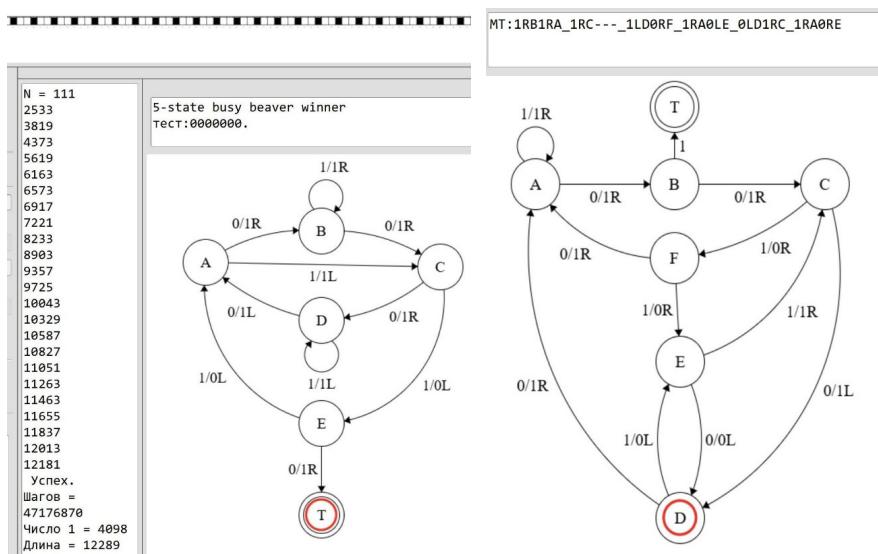


Рис. 3. Диаграммы МТ для BB(5) и кандидата BB(6)

Fig. 3. MT charts for BB(5) and candidate BB(6)

Машину BB(5) останавливается через 47176870 шагов, и это можно промоделировать в программе. На рисунке для BB(5) показано заключительное состояние. В окне конфигураций через каждые 2 миллиона шагов указывается длина текущей цепочки, на ленте — часть последней цепочки длиной 111.

Для машин BB(6) и BB(7) значения функций S и Σ настолько огромны, что для их определения найдены только нижние границы: $S(6) > \Sigma(6) > 2 > 2\uparrow\uparrow\uparrow 5$ и $S(7) > \Sigma(7) > 2\uparrow\uparrow 2\uparrow\uparrow 3$ (запись больших чисел в нотации Д. Кнута).

Доказательство BB(5) потребовало не только больших вычислительных ресурсов, но и уникальных математических методов. Для некоторых МТ было использованы инструменты автоматизации доказательств, в частности, язык Lean (Lean, 2025).



В программе имеется доступ к базе данных из 10000 машин с шестью состояниями, для которых можно ставить задачи исследования их поведения.

Один из ведущих исследователей в области теории алгоритмов, Скотт Ааронсон, так отозвался о проблеме ВВ: «... изучение «бобров» делает философские открытия Гёделя и Тьюринга «количественными и конкретными». Мы больше не говорим абстрактно о «каких-то» пределах — мы пытаемся нашупать их руками. Найти машину Тьюринга, поведение которой непредсказуемо, — это всё равно что найти первую трещину в здании общепринятой математики» (Самая сложная задача математики, 2025).

По тематике проблемы ВВ достаточно много публикаций, и знакомство с ними может заинтересовать студентов и побудить к исследованиям.

Предполагается продолжить разработку программы FSMD для реализации следующих возможностей:

- генерация случайных диаграмм с заданными характеристиками (числом вершин, дуг, алфавита символов и т.п.);
- задание и сохранение набора тестовых цепочек для автоматов;
- решение задач по аналогии с программой JFLAB;
- решения задачи построения регулярного выражения по КА;
- автоматизация удаления пустых и недостижимых состояний в КА и МТ-автоматах;
- решение задачи удаления пустых переходов;
- реализация возможности генерации отчета по задаче построения автоматов;
- построение пространственно-временных диаграмм для МТ.

В заключение приведём краткий обзор результатов использования программы FMT в учебном процессе.

В течение учебного года 2024/25 в Финансовом университете были прочитаны лекции и проведены семинарские занятия по дисциплинам «Теория компиляции» и «Теория алгоритмов». В рабочих программах этих дисциплин предусмотрено решение задач на построение конечных автоматов и машин Тьюринга. Разработанная программа оказалась очень полезным инструментом: студенты легко осваивали работу в программе и справлялись с подобными задачами. На зачетах были предложены задачи, отличные от тех, что разбирались на занятиях. Некоторые студенты, в основном те, которые пропускали занятия или не были активными, понадеялись на помощь искусственного интеллекта (ИИ). Но они просчитались: при проверке решений легко обнаруживалось, что они этим пользовались. По условию задач результат должен был представлен в виде JSON-файла, который загружался бы в программу FSMD. Но ИИ, хотя и выдавал JSON, но формат не соответствовал требованиям программы. При использовании FSMD файл формируется автоматически в требуемом формате. Кроме того, в диаграммах от ИИ присутствовали склеенные дуги и неудачное размещение вершин. При запросе к ИИ можно было бы сформулировать уточняющие детали представления решений, но это требовало дополнительных усилий со стороны студентов.

Использование ИИ студентами носит массовый характер, и этому трудно препятствовать. Решение задач в любой дисциплине способствует не только лучшему усвоению материала, но и развитию логического мышления, приобретению полезных



навыков в применении соответствующих инструментов. Студент, который вместо того, чтобы самостоятельно решать задачи, прибегает к помощи ИИ, не сможет освоить дисциплину в должной степени, получить новые знания и навыки. Поэтому правильно поставить его в условия, когда использование ИИ было бы затруднительным. Именно применение специализированных программ, подобных FSMD, позволяет это сделать. Решение задач в среде таких программ может быть обусловлено различными динамически настраиваемыми особенностями, которые ИИ не сможет учесть.

Список источников / References

1. Ахо, А., Ульман, Дж. (1978). Теория синтаксического анализа, перевода и компиляции. М.: Мир. Aho, A., & Ullman, J. (1978). *The Theory of Parsing, Translation, and Compiling*. Moscow: Mir. (In Russ.).
2. Лукин, В.Н., Чернышов, Л.Н. (2021). Вычислительные процессы: теория трансляции, управление данными и сети Петри. М.: Вузовская книга. Lukin, V.N., & Chernyshov, L.N. (2021). *Computational Processes: Translation Theory, Data Management, and Petri Nets*. Moscow: Vuzovskaya kniga. (In Russ.).
3. Поляков, В.И., Скорубский, В.И. (2012). Основы теории алгоритмов. СПб.: СПб НИУ ИТМО. Polyakov, V.I., & Skorubsky, V.I. (2012). *Fundamentals of Algorithm Theory*. St. Petersburg: ITMO University. (In Russ.).
4. Хопкрофт, Дж.Э., Мотвани, Р., Ульман, Дж. (2002). Введение в теорию автоматов, языков и вычислений (2-е изд.). М.: Издательский дом «Вильямс». Hopcroft, J.E., Motwani, R., & Ullman, J. (2002). *Introduction to Automata Theory, Languages, and Computation* (2nd ed.). Moscow: Williams Publishing House. (In Russ.).
5. Чернецкая, И.Е. (2022). Применение конечных автоматов для поиска и распознавания подстрок: методические рекомендации к лабораторной работе. Курск: Юго-Зап. гос. ун-т. URL: https://swsu.ru/sveden/files/MU_LZNo_2_Teoriya_avtomatov.pdf (дата обращения: 10.08.2025). Chernetskaya, I.E. (2022). *Application of Finite Automata for Searching and Recognizing Substrings: Methodological Guidelines for a Laboratory Work*. Kursk: Southwest State University. URL: https://swsu.ru/sveden/files/MU_LZNo_2_Teoriya_avtomatov.pdf (viewed: 10.08.2025). (In Russ.).
6. Инструмент для построения диаграммы состояния (2025). URL: <https://creately.com> (дата обращения: 10.08.2025). State diagramming tool (2025). URL: <https://creately.com> (viewed: 10.08.2025).
7. Редактор графов. URL: <https://programforyou.ru/graph-redactor> (дата обращения: 10.08.2025). Graph editor. URL: <https://programforyou.ru/graph-redactor> (viewed: 10.08.2025). (In Russ.).
8. Самая сложная задача математики? Математики ищут число, которое определит предел познания. URL: <https://www.ixbt.com/live/science/samaya-slozhnaya-zadacha-matematiki-matematiki-ischut-chislo-kotoroe-opredelit-predel-poznaniya.html> (дата обращения: 10.08.2025). The hardest problem in mathematics? Mathematicians are searching for a number that will determine the limits of knowledge. URL: <https://www.ixbt.com/live/science/samaya-slozhnaya-zadacha-matematiki-matematiki-ischut-chislo-kotoroe-opredelit-predel-poznaniya.html> (viewed: 10.08.2025). (In Russ.).



9. Эмулятор машины Тьюринга. URL: <https://programforyou.ru/calculators/turing-machine-emulator> (дата обращения: 10.08.2025).
Turing machine emulator. URL: <https://programforyou.ru/calculators/turing-machine-emulator> (viewed: 10.08.2025). (In Russ.).
10. AutomataTutor. URL: <https://automatatutor.com> (viewed: 10.08.2025).
11. Busy Beaver. URL: https://ru.wikipedia.org/wiki/Busy_beaver (дата обращения: 10.08.2025).
Busy Beaver. URL: https://ru.wikipedia.org/wiki/Busy_beaver (viewed: 10.08.2025). (In Russ.).
12. JFLAP Version 7.1 released July 27 (2018). URL: <https://www.jflap.org> (viewed: 10.08.2025).
13. Finite Automata Designer. URL: <https://fa.akdev.ir> (viewed: 10.08.2025).
14. Lean — инструмент интерактивного доказательства теорем (2025). URL: <https://ru.wikipedia.org/wiki/Lean> (дата обращения: 10.08.2025).
Lean — an interactive theorem prover (2025). URL: <https://ru.wikipedia.org/wiki/Lean> (viewed: 10.08.2025). (In Russ.).
15. The Busy Beaver Challenge (2025). URL: <https://wiki.bbchallenge.org> (viewed: 10.08.2025).
16. Turing Machine Visualization. URL: <https://turingmachine.io> (viewed: 10.08.2025).
17. Finite State Machine Designer. URL: <https://madebyevan.com> (viewed: 10.08.2025).

Информация об авторах

Лев Николаевич Чернышов, кандидат физико-математических наук, доцент, доцент, Финансовый университет при Правительстве Российской Федерации, (ФГБОУ ВО Финансовый университет), Москва, Российская Федерация, ORCID: <https://orcid.org/0000-0002-1512-4052>, e-mail: lnc Chernyshov@fa.ru

Владимир Николаевич Лукин, кандидат физико-математических наук, доцент, профессор, Московский государственный психолого-педагогический университет (ФГБОУ ВО МГППУ), Москва, Российская Федерация, ORCID: <https://orcid.org/0000-0001-8906-2686>, e-mail: lukinvn@list.ru

Information about the authors

Lev N. Chernyshov, Candidate of Physical and Mathematical Sciences, Associate Professor, Financial University under the Government of the Russian Federation, (FSBEI HE Financial University), Moscow, Russian Federation, ORCID: <https://orcid.org/0000-0002-1512-4052>, e-mail: lnc Chernyshov@fa.ru

Vladimir N. Lukin, Candidate of Physical and Mathematical Sciences, Professor, Moscow State University of Psychology and Education (MSUPE), Moscow, Russian Federation, ORCID: <https://orcid.org/0000-0001-8906-2686>, e-mail: lukinvn@list.ru

Вклад авторов

Авторы внесли равный вклад в концепцию, проведение исследования, анализ данных и подготовку рукописи.

Contribution of the authors

Authors contributed equally to the conception, conduct of the study, data analysis, and preparation of the manuscript



Чернышов Л.Н., Лукин В.Н. (2025)
Учебный тренажёр по теории алгоритмов
Моделирование и анализ данных, 2025. 15(3), 161 – 171.

Chernyshov L.N., Lukin V.N. (2025)
Training simulator for algorithm theory
Modelling and Data Analysis, 2025. 15(3), 161 – 171.

Конфликт интересов

Авторы заявляют об отсутствии конфликта интересов.

Conflict of interest

The authors declare no conflict of interest.

Поступила в редакцию 24.08.2025

Received 2025.08.24

Поступила после рецензирования 01.09.2025

Revised 2025.09.01

Принята к публикации 02.09.2025

Accepted 2025.09.02

Опубликована 30.09.2025

Published 2025.09.30