

ISSN 2949-5598

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Том 32 № 4 2025

Новосибирск
Издательство Института математики

ЗАДАЧИ БЕСКОНЕЧНОЙ РЕГУЛЯРНОЙ РЕАЛИЗУЕМОСТИ

И. Н. Шиманогов^{1, a}, *М. Н. Вялый*^{1, 2, 3, b}

¹ Московский физико-технический институт
(национальный исследовательский университет)
ул. Керченская, 1А, корп. 1, 117303 Москва, Россия

² Федеральный исследовательский центр «Информатика и управление» РАН
ул. Вавилова, 44, корп. 2, 119333 Москва, Россия

³ Национальный исследовательский университет «Высшая школа экономики»,
Покровский б-р, 11, 109028 Москва, Россия

E-mail: ^ashimanogov.in@phystech.edu, ^bvyalyi@gmail.com

Аннотация. Хорошо изученным классом алгоритмических задач являются задачи регулярной реализуемости — проверки непустоты пересечения регулярного языка с заданным языком. Такая задача имеет естественную алгебраическую интерпретацию — проверку принадлежности элемента булевой алгебры ядру определённого гомоморфизма. Это мотивирует рассмотрение аналогичной задачи бесконечной регулярной реализуемости — проверки бесконечности пересечения регулярного языка с заданным. В работе рассматриваются задачи регулярной реализуемости для разрешимых языков и приводится сравнительный анализ сложности задач бесконечной регулярной реализуемости и задач регулярной реализуемости. Библиогр. 22.

Ключевые слова: регулярный язык, задача регулярной реализуемости, арифметическая иерархия.

Введение

Одним из основных объектов изучения в теории формальных языков являются регулярные языки. Хорошо изученным классом задач являются задачи проверки выполнимости некоторого регулярного условия для слов фиксированного языка. Рассмотрим некоторый язык L . Входом задачи регулярной реализуемости является регулярный язык R , заданный, например, некоторым автоматом. Необходимо проверить непустоту пересечения L и R . Интенсивно изучался случай, когда L является

контекстно-свободным языком (см. [1–6]). Задачи регулярной реализуемости для контекстно-свободных языков оказываются важными для теории тонких сводимостей (fine-grained reductions) (см. [3, 6]). В [7] рассмотрены задачи регулярной реализуемости, связанные с комбинаторикой слов. Разрешимость задач регулярной реализуемости для широкого круга теоретико-графовых задач в подходящей кодировке графов установлена в работах [8–10]. С использованием аналогичной кодировки в [11] доказана разрешимость задач регулярной реализуемости, связанных с целочисленным программированием.

Общая задача регулярной реализуемости введена М. Вялым в 2009 г. Основной мотивацией была надежда найти ограничения на возможные уровни алгоритмической сложности по аналогии со знаменитой гипотезой CSP (для задач CSP выполняется дихотомия: либо задача принадлежит P, либо NP-полна; точную исходную формулировку см. в [12]). Позднее гипотеза CSP была доказана в 2017 г. независимо А. Булатовым [13] и Д. Жуком [14], подробное изложение этих доказательств см. в [15, 16]. Однако для задач регулярной реализуемости ситуация принципиально иная. В 2013 г. М. Вялый доказал универсальность задач регулярной реализуемости относительно алгоритмической сложности с точностью до некоторых достаточно слабых сводимостей. Позднее была доказана универсальность нескольких ограниченных классов задач регулярной реализуемости (см. [17]). Там же можно найти обзор более ранних результатов про универсальность задач регулярной реализуемости. Заметим, что самый популярный случай задач регулярной реализуемости для контекстно-свободных языков остаётся открытым, причём результаты [2] показывают, что и в этом случае разнообразие возможных уровней алгоритмической сложности может быть весьма велико.

Каждому языку L соответствует булева алгебра относительно регулярных языков: данная алгебра образована всеми возможными пересечениями регулярных языков с L с операциями пересечения, объединения и дополнения до L . При этом существует естественный гомоморфизм из алгебры регулярных языков на относительно регулярную алгебру L : $\varphi(R) = R \cap L$. Таким образом, задача регулярной реализуемости является задачей проверки того, что образ данного элемента не равен нулю.

В данной работе рассматриваем следующую, схожую задачу — алгоритмическую проверку того, что образ данного элемента не лежит в идеале Фреше, т. е. не является конечным объединением атомов. Так как это равносильно бесконечности пересечения L и R , данные задачи названы задачами бесконечной регулярной реализуемости. В заключительной части работы более подробно обсуждаются интерпретации доказанных утверждений с точки зрения булевых алгебр.

В работе рассматриваются задачи регулярной реализуемости для разрешимых языков и приводится анализ алгоритмической сложности задачи бесконечной регулярной реализуемости. Доказывается существование разрешимых языков таких, что задача регулярной реализуемости разрешима, а бесконечной регулярной реализуемости — нет, и наоборот. Так же приводится пример языка, когда данные задачи полны в Σ_1 и Π_2 соответственно.

1. Кодировка машин Тьюринга

Так как множество машин Тьюринга (МТ) счётно, каждой из них возможно сопоставить некоторое натуральное число. Стандартная кодировка строится следующим образом: через разделитель записывается количество состояний в машине, размер её алфавита, номера особых состояний, особых символов и все переходы в виде кортежей из номеров элементов упомянутых множеств. Далее получившееся слово приводится, например, к алфавиту $\{0, 1\}$ и трактуется как натуральное число.

Нумерацией назовём функцию, сопоставляющую каждому натуральному числу некоторую МТ. В дальнейшем нам понадобится особая нумерация машин Тьюринга со следующим свойством: для любой бесконечной арифметической прогрессии в этой нумерации для любой машины Тьюринга в выбранной прогрессии имеется бесконечное множество чисел, которым сопоставлена указанная МТ. При этом необходимо, чтобы стандартная кодировка была вычислима по данному особому номеру.

Последовательно будем строить частично определённые функции f_i с конечными областями определения, сопоставляющие натуральным числам номера машин Тьюринга в стандартной кодировке.

Пронумеруем все бесконечные арифметические прогрессии вычислимым образом. Пусть отображение на функциях $A_i^j(f)$ определено следующим образом: выделим среди элементов, на которых не определена f , наименьший, принадлежащий прогрессии под номером i , и зададим на нём значение j . Через B_i^j обозначим следующую композицию отображений: $A_i^j \circ A_{i-1}^j \circ \dots \circ A_1^j$. Таким образом, применение к функции f отображения B_i^j гарантирует, что для любой из первых i арифметических прогрессий найдётся число n из этой прогрессии такое, что $B_i^j[f](n) = j$.

Рассмотрим композицию $C_i = B_i^i \circ B_{i-1}^{i-1} \circ \dots \circ B_1^1$. Применение к функции f отображения C_i гарантирует, что для любой из первых i арифметических прогрессий найдутся числа n_1, \dots, n_i из этой прогрессии такие, что $C_i[f](n_i) = i$. Будем считать, что f_0 нигде не определена. Положим $f_i = C_i(f_{i-1})$ и $f = \bigcup f_i$.

Заметим, что такая функция f вычислима и всюду определена. Действительно, $f(n)$ определяется за конечное число шагов C_i , поскольку A_i^j

всегда увеличивает область определения функции таким образом, что она остаётся начальным отрезком натуральных чисел. Так как A_i^j всегда только доопределяет функцию в точке, где она ещё не определена, определение f как функции корректно.

Теперь рассмотрим следующую нумерацию МТ: номер n соответствует МТ со стандартным номером $f(n)$. При этом для каждой арифметической прогрессии i и для каждого номера j операция A_i^j входит во все C_k , где $k = \max(i, j)$. Таким образом, в любой арифметической прогрессии для любой машины встречается бесконечно много чисел, нумерующих данную машину.

Всюду далее используется построенная кодировка.

2. Арифметическая иерархия

Будем рассматривать предикаты как функции в множество $\{0, 1\}$.

Определение 1. Язык L будем называть *разрешимым*, если предикат принадлежности данному языку вычислим. Обозначим множество разрешимых языков через $\Delta_0 = \Sigma_0 = \Pi_0$.

Определение 2. Для $n \geq 1$ будем говорить, что

- L принадлежит Σ_n , если

$$x \in L \Leftrightarrow (\exists y_1)(\forall y_2)(\exists y_3) \dots (Qy_n) R(x, y_1, y_2, \dots, y_n),$$

где Q — это \exists для нечётного n и \forall для чётного, а R — вычислимый предикат;

- L принадлежит Π_n , если

$$x \in L \Leftrightarrow (\forall y_1)(\exists y_2)(\forall y_3) \dots (Qy_n) R(x, y_1, y_2, \dots, y_n),$$

где Q — это \forall для нечётного n и \exists для чётного, а R — вычислимый предикат;

- L принадлежит Δ_n , если $L \in \Pi_n$ и $L \in \Sigma_n$, т. е. $\Delta_n = \Pi_n \cap \Sigma_n$.

Теорема 1 [18]. Для классов Σ_n , Π_n , Δ_n выполняются следующие соотношения:

- 1) $A \in \Sigma_n \Leftrightarrow \bar{A} \in \Pi_n$;
- 2) $A \in \Sigma_n \cup \Pi_n \Rightarrow (\forall m > n) A \in \Sigma_m \cap \Pi_m$;
- 3) $A, B \in \Sigma_n \Rightarrow A \cup B, A \cap B \in \Sigma_n$;
- 4) $A, B \in \Pi_n \Rightarrow A \cup B, A \cap B \in \Pi_n$;
- 5) $R \in \Sigma_n \wedge A = \{x \mid \exists y: (x, y) \in R\} \Rightarrow A \in \Sigma_n$.

Теорема 2 (теорема Поста (см. [18])).

$$\Delta_1 = \Delta_0.$$

Определение 3. Будем говорить, что язык A сводится к языку B , и используем обозначение $A \leq B$, если существует вычислимая функция f такая, что $x \in A$ тогда и только тогда, когда $f(x) \in B$.

Определение 4. Будем говорить, что язык A труден в классе Γ , если для любого $B \in \Gamma$ имеет место $B \leq A$. Если, кроме того, $A \in \Gamma$, то будем говорить, что A полон в Γ .

Лемма 1 (см. [18]). Пусть $A \leq B$. Тогда

- 1) если $B \in \Sigma_n$, то $A \in \Sigma_n$;
- 2) если $B \in \Pi_n$, то $A \in \Pi_n$;
- 3) если A труден в Σ_n , то B труден в Σ_n ;
- 4) если A труден в Π_n , то B труден в Π_n .

Лемма 2 (см. [18]). Имеют место следующие утверждения:

- 1) если A полон в Σ_n , то \overline{A} полон в Π_n ;
- 2) если A полон в Π_n , то \overline{A} полон в Σ_n .

Лемма 3. Следующий язык разрешим:

$$\text{Time} = \{(x, y, t) \mid \text{MT с номером } x \text{ на слове } y \\ \text{останавливается на шаге } t\}.$$

ДОКАЗАТЕЛЬСТВО. Достаточно воспользоваться конструкцией универсальной машины Тьюринга из [18] и на отдельной ленте хранить число сделанных машиной x шагов, после чего, как только оно станет равно t , дать ответ. Лемма 3 доказана.

Лемма 4 (см. [18]). Следующий язык полон в Σ_1 :

$$\text{Halt} = \{(x, y) \mid \text{MT с номером } x \text{ останавливается на слове } y\}.$$

Лемма 5 (см. [18]). Следующий язык полон в Σ_1 :

$$\text{Halt}_\varepsilon = \{x \mid \text{MT с номером } x \text{ останавливается на пустом слове } \varepsilon\}.$$

Лемма 6 (см. [18]). Следующий язык полон в Π_2 :

$$\text{Inf} = \{x \mid \text{MT с номером } x \text{ останавливается} \\ \text{на бесконечном множестве слов}\}.$$

3. Регулярные языки

Будем считать, что регулярные языки и автоматы определены в соответствии с [19]. Детерминированный и недетерминированный конечные автоматы кратко именуем ДКА и НКА соответственно.

Пусть дан автомат A . Будем обозначать распознаваемый им язык через $\mathcal{L}(A)$. Приведём теорему, дающую характеристику регулярных языков в унарном алфавите.

Теорема 3 (см. [20]). *Унарный язык регулярен тогда и только тогда, когда множество длин его слов является конечным объединением арифметических прогрессий.*

Выделим важное свойство построенной кодировки машин Тьюринга.

Лемма 7. *В любом бесконечном унарном регулярном языке R есть бесконечно много номеров любой машины Тьюринга.*

ДОКАЗАТЕЛЬСТВО следует из теоремы 3 и свойств построенной нумерации машин. Лемма 7 доказана.

Сформулируем также несколько вспомогательных утверждений о вычислимости некоторых операций над языками. Будем считать, что ДКА на вход алгоритму подаётся в виде стандартного описания.

Лемма 8. *Функция, принимающая на вход два ДКА и возвращающая ДКА для пересечения языков входных автоматов, вычислима.*

ДОКАЗАТЕЛЬСТВО. Автомат для пересечения языков строится с использованием конструкции автоматов, описанной например в [19]. Лемма 8 доказана.

Лемма 9. *Функция, принимающая на вход регулярное выражение и возвращающая эквивалентный ДКА, вычислима.*

ДОКАЗАТЕЛЬСТВО. Алгоритм строится непосредственно из доказательства теоремы Клини об эквивалентности классов языков, задаваемых регулярными выражениями и ДКА (см. [19]). Лемма 9 доказана.

Лемма 10. *Функция, принимающая на вход ДКА M и возвращающая автомат, распознающий все префиксы слов языка $\mathcal{L}(M)$, вычислима.*

ДОКАЗАТЕЛЬСТВО. Действительно, достаточно сделать финальными все состояния автомата M , из которых достижимо какое-либо финальное состояние M . Лемма 10 доказана.

Лемма 11. *Язык ДКА, распознающих конечные языки, разрешим.*

ДОКАЗАТЕЛЬСТВО. Отметим, что задача достижимости на графе автомата разрешима, например при помощи обхода данного графа в глубину. Для каждого состояния, из которого достижимо финальное и которое достижимо из стартового, проверим, достижимо ли оно само из себя. Язык бесконечен тогда и только тогда, когда в нём есть слова сколь угодно большой длины, что равносильно тому, что существует такой путь, на котором есть состояние, достижимое из самого себя. Таким образом, язык бесконечен тогда и только тогда, когда состояние с указанным свойством найдено. Лемма 11 доказана.

Лемма 12. *Язык ДКА, распознающих пустой язык, разрешим.*

Доказательство. Проверим, существует ли такое состояние, которое достижимо из стартового и из которого достижимо финальное состояние. Язык непуст тогда и только тогда, когда такое состояние существует. Лемма 12 доказана.

Лемма 13. *Если A — ДКА и $\mathcal{L}(A) \subset 0^*1^*$, то вычислима функция, отображающая A в конечное множество ДКА $\{Q_1, \dots, Q_n, P_1, \dots, P_n\}$, где*

- 1) $\mathcal{L}(Q_i) \subset 0^*$;
- 2) $\mathcal{L}(P_i) \subset 1^*$;
- 3) $\mathcal{L}(A) = \bigcup_{i=1}^n \mathcal{L}(Q_i)\mathcal{L}(P_i)$.

Доказательство. Для произвольного состояния i автомата A построим автомат Q_i для всех путей из стартового состояния в i , состоящих только из символа 0, и автомат P_i для всех путей из i в финальное состояние, состоящих только из символа 1. Оставим только такие пары автоматов Q_i, P_i , языки которых непусты. Тогда действительно $\mathcal{L}(A) = \bigcup_{i=1}^n \mathcal{L}(Q_i)\mathcal{L}(P_i)$. Лемма 13 доказана.

4. Задачи регулярной реализуемости

Зачастую (см., например, [17]) при рассмотрении задач регулярной реализуемости входом считается некоторый НКА. В дальнейшем будем считать, что на вход подаётся ДКА. Так как построение эквивалентного ДКА — вычислимая операция (см. [19]), это не меняет доказанных свойств.

Определение 5. *Задачей регулярной реализуемости для языка L назовём язык*

$$\text{DRR}_L = \{A \mid |L \cap \mathcal{L}(A)| \neq \emptyset\}.$$

Лемма 14. *Если язык L разрешим, то $\text{DRR}_L \in \Sigma_1$.*

Доказательство. Пересечение L с входным языком непусто тогда и только тогда, когда существует слово, принадлежащее им обоим, иначе говоря,

$$A \in \text{DRR}_L \Leftrightarrow \exists w: A(w) \wedge (w \in L).$$

Если L разрешим, то под кванторами находится разрешимый предикат, следовательно, $\text{DRR}_L \in \Sigma_1$. Лемма 14 доказана.

5. Задачи бесконечной регулярной реализуемости

Определение 6. Задачей бесконечной регулярной реализуемости для языка L назовём язык

$$\text{DRR}_L^\infty = \{A \mid |L \cap \mathcal{L}(A)| = \infty\}.$$

Лемма 15. Если язык L разрешим, то $\text{DRR}_L^\infty \in \Pi_2$.

Доказательство. Пересечение L с входным языком бесконечно тогда и только тогда, когда в нём есть слово произвольной длины, иначе говоря,

$$A \in \text{DRR}_L^\infty \Leftrightarrow \forall v \exists w: A(w) \wedge (w \in L) \wedge (|w| > |v|).$$

Если L разрешим, то под кванторами находится разрешимый предикат, следовательно, $\text{DRR}_L^\infty \in \Pi_2$. Лемма 15 доказана.

Лемма 16. Для любого языка L задача DRR_{0L}^∞ сводится к DRR_L^∞ .

Доказательство. Рассмотрим сводящую функцию f , которая по автомату A строит автомат для языка $\{x \mid 0x \in \mathcal{L}(A)\}$. Такая функция вычислима, так как для построения соответствующего автомата достаточно определить, какие состояния достижимы из стартового по 0. Тогда имеет место цепочка эквивалентностей

$$A \in \text{DRR}_{0L}^\infty \Leftrightarrow |0L \cap \mathcal{L}(A)| = \infty \Leftrightarrow |L \cap \mathcal{L}(f(A))| = \infty \Leftrightarrow f(A) \in \text{DRR}_L^\infty.$$

Лемма 16 доказана.

6. Основные результаты

Лемма 17. Если языки L и DRR_L разрешимы, то $\text{DRR}_L^\infty \in \Pi_1$.

Доказательство. Рассмотрим ДКА A и положим

$$M_{i+} = \{w \in \mathcal{L}(A) \mid |w| \geq i\}.$$

Построим автомат A_{i+} для языка M_{i+} как пересечение автомата A и автомата для языка $\Sigma^i \Sigma^*$.

Обозначим через $t(A, i)$ функцию, отображающую автомат A в автомат A_{i+} . Эта функция вычислима в силу лемм 8 и 9. Пересечение произвольного языка с L бесконечно тогда и только тогда, когда для любого i непусто пересечение подмножества его слов длины не меньше i с языком L . Другими словами,

$$A \in \text{DRR}_L^\infty \Leftrightarrow \forall v t(A, |v|) \in \text{DRR}_L.$$

Если задача регулярной реализуемости для языка L разрешима, то под кванторами находится разрешимый предикат, следовательно, $\text{DRR}_L^\infty \in \Pi_1$.

Заметим, что требование разрешимости L избыточно, так как оно следует из разрешимости DRR_L . Действительно, вычислимая функция f , сопоставляющая слову w автомат с языком $\{w\}$, является сводимостью L к DRR_L . Лемма 17 доказана.

Теорема 4. *Существует разрешимый язык L такой, что DRR_L разрешим, а DRR_L^∞ полон в Π_1 .*

ДОКАЗАТЕЛЬСТВО. Рассмотрим язык

$$L = \{0^x 1^t \mid \text{МТ с номером } x \text{ не останавливается на } \varepsilon \text{ за } t \text{ шагов}\}.$$

Язык L сводится к $\overline{\text{Time}}$ с помощью функции $f(0^x 1^t) = (x, \varepsilon, t)$, поэтому разрешим по леммам 1 и 3 и теореме 1.

Сначала покажем разрешимость DRR_L . Пусть на вход подан автомат A , положим $R = \mathcal{L}(A)$. Построим автомат для языка $R' = R \cap 0^* 1^*$. Эта операция вычислима в силу лемм 8 и 9. Заметим, что $R \cap L = R' \cap L$, поэтому если $R' = \emptyset$, то $R \cap L = \emptyset$, а значит, $A \notin \text{DRR}_L$. Проверка R' на пустоту является вычислимой операцией в силу леммы 12.

Пусть R' непуст. Построим по нему множество автоматов $\{Q_1, \dots, Q_n, P_1, \dots, P_n\}$, как описано в лемме 13. Язык R' пересекается с L тогда и только тогда, когда какой-то $\mathcal{L}(Q_i)\mathcal{L}(P_i)$ пересекается с L . Рассмотрим каждую пару автоматов Q_i, P_i в отдельности. Возможны несколько случаев в зависимости от конечности их языков.

Если $\mathcal{L}(Q_i)$ бесконечен, то имеем бесконечный унарный язык и по лемме 7 в нём встречается слово, длина которого соответствует номеру МТ, не останавливающейся за любое число шагов, а значит, пересечение $L \cap \mathcal{L}(Q_i)\mathcal{L}(P_i) \neq \emptyset$ и $R \in \text{DRR}_L$.

Если язык $\mathcal{L}(Q_i)$ конечен, то обратим внимание на автомат P_i . В случае конечного $\mathcal{L}(P_i)$ проверим каждое слово $0^q 1^p \in \mathcal{L}(Q_i)\mathcal{L}(P_i)$. Пересечение $L \cap \mathcal{L}(Q_i)\mathcal{L}(P_i)$ непусто тогда и только тогда, когда МТ с номером q не останавливается за p шагов на пустом слове, а это вычислимая операция, так как разрешим язык Time .

В случае бесконечного $\mathcal{L}(P_i)$ (и конечного $\mathcal{L}(Q_i)$) рассмотрим каждое слово $0^q \in \mathcal{L}(Q_i)$. Выберем наименьшее по длине слово $1^p \in \mathcal{L}(P_i)$ и проверим, останавливается ли МТ с номером q на пустом слове за p шагов. Если останавливается, то пересечение $L \cap \mathcal{L}(Q_i)\mathcal{L}(P_i)$ пусто, иначе нет.

Теперь покажем, что DRR_L^∞ полон в Π_1 . Построим сводимость $\overline{\text{HalT}}_\varepsilon$ к L и воспользуемся леммами 1, 2 и 5. Заметим, что функция $f(x) = A$, где $\mathcal{L}(A) = (0^x 1^*)$, вычислима в силу леммы 9. Действительно, $0^x 1^*$ имеет бесконечное пересечение с L тогда и только тогда, когда МТ с номером x не остановится на пустом слове. Тогда в силу лемм 1, 2 и 5 верно, что DRR_L^∞ полон в Π_1 . Теорема 4 доказана.

Теорема 5. *Существует разрешимый язык L такой, что DRR_L^∞ разрешим, а DRR_L полон в Σ_1 .*

Доказательство. Рассмотрим язык

$$L = \{0^x 1^t \mid \text{МТ с номером } x \text{ останавливается на } \varepsilon \text{ на шаге } t\}.$$

Язык L сводится к **Time** с помощью функции $f(0^x 1^t) = (x, \varepsilon, t)$, поэтому разрешим по леммам 1 и 3.

Сначала покажем разрешимость DRR_L^∞ . Пусть на вход подан автомат A , обозначим $R = \mathcal{L}(A)$. Построим автомат для языка $R' = R \cap 0^* 1^*$. Эта операция вычислима в силу лемм 8 и 9. Заметим, что $R \cap L = R' \cap L$, поэтому если R' конечен, то конечно и $R \cap L$, а значит, $A \notin \text{DRR}_L^\infty$. Проверка R' на конечность является вычислимой операцией в силу леммы 11.

Пусть R' бесконечен. Рассмотрим тогда R_0 — язык, получаемый пересечением языка префиксов R' с 0^* (построение автомата для такого языка вычислимо в силу лемм 8–10). Если R_0 конечен, то конечно и пересечение R' и L , так как в L для каждого i лежит не более одного слова из $0^i 1^*$.

Пусть R_0 бесконечен. Тогда существует состояние q автомата для языка R' такое, что из стартового в q ведёт бесконечно много слов из 0^* и есть слово 1^z , которое ведёт из q в финальное состояние. Однако в унарном регулярном языке слов, которые ведут из стартового состояния в q , бесконечно много раз встречается номер любой МТ по лемме 7. В том числе и тех МТ, которые останавливаются за z шагов. Следовательно, пересечение $R \cap L$ бесконечно, и $A \in \text{DRR}_L^\infty$.

Теперь покажем, что DRR_L полно в Σ_1 . Построим сводимость Halt_ε к DRR_L с помощью функции $f(x) = A$, где $\mathcal{L}(A) = 0^x 1^*$, которая вычислима в силу леммы 9. Действительно, $0^x 1^*$ пересекается с L тогда и только тогда, когда МТ с номером x остановится на пустом слове. Тогда в силу лемм 1 и 5 верно, что DRR_L полон в Σ_1 . Теорема 5 доказана.

Теорема 6. *Существует разрешимый язык L такой, что DRR_L полон в Σ_1 , а DRR_L^∞ полон в Π_2 .*

Доказательство. Зададим на словах нумерацию в алфавитном порядке (пусть слово w_y имеет номер y) и рассмотрим язык

$$L = \{0^x 1^y 2^t \mid \text{МТ с номером } x \text{ останавливается на слове } w_y \text{ на шаге } t\}.$$

Язык L сводится к **Time** с помощью функции $f(0^x 1^y 2^t) = (x, w_y, t)$, поэтому разрешим по леммам 1 и 3. В силу разрешимости L из лемм 14 и 15 следует, что $\text{DRR}_L \in \Sigma_1$, а $\text{DRR}_L^\infty \in \Pi_2$.

Для начала докажем, что DRR_L труден в Σ_1 , построив сводимость Halt к DRR_L . Рассмотрим вычислимую (по лемме 9) функцию $f: (x, w_y) \rightarrow A$, каждой паре из номера МТ и слова сопоставляющую ДКА A такой, что

$\mathcal{L}(A) = 0^x 1^y 2^*$. Если данная МТ останавливалась на данном слове (иначе говоря, $(x, w_y) \in \text{Halt}$), то существует шаг p , на котором она останавливается. Тогда $0^x 1^y 2^*$ пересекается с L , так как в них обоих лежит слово $0^x 1^y 2^p$, а значит, $f(x, w_y) \in \text{DRR}_L$. Обратно, если МТ x не останавливается на слове y , то никакое слово из $0^x 1^y 2^*$ не лежит в L , следовательно, $f(x, y) \notin \text{DRR}_L$.

Докажем, что DRR_L^∞ труден в Π_2 , построим сводимость Inf к DRR_L^∞ . Рассмотрим вычислимую (по лемме 9) функцию $f: x \rightarrow A$, которая каждому номеру МТ сопоставляет ДКА A такой, что $\mathcal{L}(A) = 0^x 1^* 2^*$. Данная МТ останавливается на конкретном слове w_y тогда и только тогда, когда существует шаг p , на котором она останавливается. Тогда $0^x 1^y 2^*$ пересекается с L , так как в них обоих лежит слово $0^x 1^y 2^p$, при этом мощность данного пересечения равна 1. Значит, бесконечность множества слов, на которых останавливается МТ с номером x , равносильна бесконечности пересечения L с $0^x 1^* 2^*$. Теорема 6 доказана.

Теорема 7. *Существует разрешимый язык L такой, что DRR_L полон в Σ_1 , а DRR_L^∞ полон в Π_1 .*

Доказательство. Обозначим язык из теоремы 5 через L_0 , а язык из теоремы 4 через L_1 . Рассмотрим язык $L = 0L_0 \cup 1L_1$. Он разрешим, так как разрешимы L_0 и L_1 .

Из леммы 14 следует, что $\text{DRR}_L \in \Sigma_1$. Покажем, что $\text{DRR}_L^\infty \in \Pi_1$. По определению $A \in \text{DRR}_L^\infty$ равносильно тому, что $|0L_0 \cap \mathcal{L}(A)| = \infty$ или $|1L_1 \cap \mathcal{L}(A)| = \infty$. Следовательно, $\text{DRR}_L^\infty = \text{DRR}_{0L_0}^\infty \cup \text{DRR}_{1L_1}^\infty$. По теореме 1 и леммам 1, 16 получаем, что $\text{DRR}_L^\infty \in \Pi_1$.

Покажем, что DRR_L труден в Σ_1 . Для этого построим функцию f , сводящую DRR_L к DRR_{L_0} . Пусть f отображает A в автомат для языка $0\mathcal{L}(A)$. Так как это конкатенация регулярных языков, f вычислима, при этом

$$\begin{aligned} A \in \text{DRR}_L &\Leftrightarrow |L \cap \mathcal{L}(A)| \neq \emptyset \Leftrightarrow \\ &\Leftrightarrow |0L_0 \cap \mathcal{L}(A)| \neq \emptyset \vee |1L_1 \cap \mathcal{L}(A)| \neq \emptyset \Leftrightarrow \\ &\Leftrightarrow |0L_0 \cap \mathcal{L}(f(A))| \neq \emptyset \Leftrightarrow f(A) \in \text{DRR}_{L_0}. \end{aligned}$$

Осталось показать, что DRR_L^∞ труден в Π_1 . Для этого построим функцию f , сводящую DRR_L^∞ к $\text{DRR}_{L_1}^\infty$. Пусть f отображает A в автомат для языка $1\mathcal{L}(A)$. Так как это конкатенация регулярных языков, f вычислима, при этом

$$\begin{aligned} A \in \text{DRR}_L^\infty &\Leftrightarrow |L \cap \mathcal{L}(A)| = \infty \Leftrightarrow \\ &\Leftrightarrow |0L_0 \cap \mathcal{L}(A)| = \infty \vee |1L_1 \cap \mathcal{L}(A)| = \infty \Leftrightarrow \\ &\Leftrightarrow |1L_1 \cap \mathcal{L}(f(A))| = \infty \Leftrightarrow f(A) \in \text{DRR}_{L_1}^\infty. \end{aligned}$$

Теорема 7 доказана.

Заметим, что в случае двухбуквенного алфавита приведённые результаты сохраняются. Действительно, рассмотрим отображение $f: 0 \mapsto 00, 1 \mapsto 11, 2 \mapsto 01$, которое естественным образом продолжается до отображения на языках. Отображение f инъективно, и обратное к нему также инъективно, хотя и не всюду определено. Остаётся заметить, что вычислима функция, сопоставляющая автомату A автомат B такой, что $\mathcal{L}(B) = f(\mathcal{L}(A))$. Действительно, достаточно, например, построить по автомату регулярное выражение, подействовать на все символы в нём отображением f , после чего построить автомат по регулярному выражению. Случай однобуквенного алфавита является вопросом для дальнейшего изучения.

7. Относительно регулярные алгебры

Так как булевы алгебры, состоящие из пересечений заданного языка с регулярными, образуют довольно богатый класс (а именно, любая атомная булева алгебра изоморфна некоторой в этом классе [21]), отдельный интерес вызывает приложение задач регулярной реализуемости к изучению их свойств. В данном разделе определения даются в соответствии с [22].

Определение 7. *Булевой алгеброй* называется множество M с бинарными алгебраическими операциями \vee и \wedge , а также унарной алгебраической операцией \neg такими, что для любых $a, b, c \in M$ имеют место равенства

- (1) $a \vee b = b \vee a$;
- (2) $a \vee (b \vee c) = (a \vee b) \vee c$;
- (3) $a \vee a = a$;
- (4) $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$;
- (5) $\neg(a \vee b) = \neg a \wedge \neg b$;
- (6) $(a \wedge \neg a) \vee b = b$;
- (7) $\neg\neg a = a$.

На произвольной булевой алгебре можно ввести отношение частичного порядка: $a \leq b \Leftrightarrow a \wedge b = a$. Минимальный и максимальный элементы обозначим через 0 и 1 соответственно. Нетрудно проверить, что в булевой алгебре такие элементы уникальны и обладают свойствами $a \vee 0 = a = a \wedge 1$, $a \wedge 0 = 0$ и $a \vee 1 = 1$. Особый интерес при изучении булевых алгебр играют атомы.

Определение 8. Ненулевой элемент a будем называть *атомом*, если из отношения $b \leq a$ следует, что $b = 0$ или $b = a$.

Далее будем иметь дело только со счётными атомными булевыми алгебрами.

Определение 9. Булева алгебра называется *счётной*, если мощность счётна.

Определение 10. Алгебру A будем называть *атомной*, если для любого элемента $a \in A$ существует атом $b \in A$ такой, что $b \leq a$.

Так как регулярные языки замкнуты относительно объединения, пересечения и дополнения, они образуют булеву алгебру REG.

Определение 11. *Относительно регулярной алгеброй* для языка L будем называть булеву алгебру $\text{REG}_L = \{M \cap L \mid M \in \text{REG}\}$ с операциями объединения, пересечения и дополнения до L .

Заметим, что REG_{Σ^*} совпадает с REG. Из определения также видно, что относительно регулярная алгебра всегда является гомоморфным образом алгебры регулярных языков, а соответствующий гомоморфизм имеет вид $\varphi(x) = x \cap L$. Стоит отметить, что относительно регулярные алгебры всегда атомны: в роли атомов в них выступают языки из одного слова.

Как и для других алгебраических структур, важным оказывается понятие идеала булевой алгебры, особую роль в изучении булевых алгебр играет идеал Фреше.

Определение 12. Подмножество $I \subset A$ булевой алгебры A называется *идеалом*, если I содержит 0, для каждого элемента i из I все меньшие i элементы также содержатся в I и I замкнуто относительно операции \vee .

Отдельно подчеркнём, что идеал булевой алгебры сам может не являться булевой алгеброй.

Определение 13. *Идеалом Фреше* $F(A)$ булевой алгебры A называется идеал, состоящий из всех конечных дизъюнкций её атомов.

В относительно регулярной алгебре для языка L есть простой способ указать на её элемент X : достаточно указать регулярный язык M , пересечение которого с L равно X . Такое представление неоднозначно, что создаёт дополнительные проблемы в алгоритмических вопросах о данной булевой алгебре. Задача проверки равенства двух элементов сводится к задаче DRR_L для их симметрической разности (это также регулярный язык). Если интересоваться фактором относительно регулярной алгебры по идеалу Фреше, то проверка равенства двух элементов сводится к задаче бесконечной регулярной реализуемости для их симметрической разности (при представлении элементов этой фактор-алгебры регулярными языками). Такая связь между задачами регулярной реализуемости и алгоритмическими вопросами о булевых алгебрах представляется нам важной и заслуживающей дальнейшего изучения.

Авторы благодарны рецензенту за полезные замечания и предложения, существенно улучшившие содержание работы.

Финансирование работы

Работа второго автора выполнена в рамках Программы фундаментальных исследований НИУ ВШЭ, а также имеет частичную финансовую поддержку в рамках гос. задания (проект № FFNG-2024-0003). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. **Bouajjani A., Esparza J., Maler O.** Reachability analysis of pushdown automata: Application to model-checking // CONCUR'97: Concurrency theory. Proc. 8th Int. Conf. (Warsaw, Poland, July 1–4, 1997). Heidelberg: Springer, 1997. P. 135–150. (Lect. Notes Comput. Sci.; V. 1243). DOI: 10.1007/3-540-63141-0_10.
2. **Rubtsov A., Vyalyi M.** Regular realizability problems and context-free languages // Descriptive complexity of formal systems. Proc. 17th Int. Workshop DCFS 2015 (Waterloo, ON, Canada, June 25–27, 2015). Cham: Springer, 2015. P. 256–267. (Lect. Notes Comput. Sci.; V. 9118). DOI: 10.1007/978-3-319-19225-3_22.
3. **Chistikov D., Majumdar R., Schepper P.** Subcubic certificates for CFL reachability // Proc. ACM Program. Lang. 2022. V. 6, No. POPL. Article ID 41. 29 p. DOI: 10.1145/3498702.
4. **Pavlogiannis A.** CFL/Dyck reachability: An algorithmic perspective // ACM SIGLOG News. 2023. V. 9, No. 4. P. 5–25. DOI: 10.1145/3583660.3583664.
5. **Kjelstrøm A. H., Pavlogiannis A.** The decidability and complexity of interleaved bidirected Dyck reachability // Proc. ACM Program. Lang. 2022. V. 6, No. POPL. Article ID 12. 26 p. DOI: 10.1145/3498673.
6. **Koutris P., Deep S.** The fine-grained complexity of CFL reachability // Proc. ACM Program. Lang. 2023. V. 7, No. POPL. Article ID 59. 27 p. DOI: 10.1145/3571252.
7. **Anderson T., Loftus J., Rampersad N., Santean N., Shallit J.** Detecting palindromes, patterns and borders in regular languages // Inf. Comput. 2009. V. 207. P. 1096–1118. DOI: 10.1016/j.ic.2008.06.007.
8. **Wolf P., Fernau H.** Regular intersection emptiness of graph problems: Finding a needle in a haystack of graphs with the help of automata. Ithaca, NY, 2020. 29 p. (e-Print Archive / Cornell Univ.; arXiv:2003.05826). DOI: 10.48550/arXiv.2003.05826.
9. **Wolf P.** From decidability to undecidability by considering regular sets of instances // Theor. Comput. Sci. 2022. V. 899. P. 25–38. DOI: 10.1016/j.tcs.2021.11.006.

10. **Diekert V., Fernau H., Wolf P.** Properties of graphs specified by a regular language // Acta Inform. 2022. V. 59. P. 357–385. DOI: 10.1007/s00236-022-00427-z.
11. **Wolf P.** On the decidability of finding a positive ILP-instance in a regular set of ILP-instances // Acta Inform. 2022. V. 59. P. 505–519. DOI: 10.1007/s00236-022-00429-x.
12. **Feder T., Vardi M. Y.** The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory // SIAM J. Comput. 1999. V. 28, No. 1. P. 57–104.
13. **Bulatov A. A.** A dichotomy theorem for nonuniform CSPs // Proc. 58th IEEE Annu. Symp. Foundations of Computer Science (Berkeley, CA, USA, Oct. 15–17, 2017). Piscataway: IEEE, 2017. P. 319–330. DOI: 10.1109/FOCS.2017.37
14. **Zhuk D.** A proof of CSP dichotomy conjecture // Proc. 58th IEEE Annu. Symp. Foundations of Computer Science (Berkeley, CA, USA, Oct. 15–17, 2017). Piscataway: IEEE, 2017. P. 331–342. DOI: 10.1109/FOCS.2017.38
15. **Bulatov A. A.** A dichotomy theorem for nonuniform CSPs. Ithaca, NY, 2017. 101 p. (e-Print Archive / Cornell Univ.; arXiv:1703.03021). DOI: 10.48550/arXiv.1703.03021.
16. **Zhuk D.** A proof of the CSP dichotomy conjecture // J. ACM. 2020. V. 67, No. 5. Article ID 30. 78 p. DOI: 10.1145/3402029.
17. **Вялый М. Н., Рубцов А. А.** Задачи регулярной реализуемости для описаний конечных отношений // Пробл. передачи информации. 2024. Т. 60, № 3. С. 46–58.
18. **Soare R.** Turing computability: Theory and applications. Heidelberg: Springer, 2016. 264 p. DOI: 10.1007/978-3-642-3.
19. **Kozen D.** Automata and computability. New York: Springer, 2012. 400 p. DOI: 10.1007/978-1-4612-1844-9.
20. **Shallit J. O.** A second course in formal languages and automata theory. New York: Camb. Univ. Press, 2008. 240 p. DOI: 10.1017/SB09780511808876.
21. **Шиманогов И. Н., Вялый М. Н.** Классификация относительно регулярных алгебр // Тр. МФТИ. 2024. Т. 16, № 4. С. 128–134.
22. **Гончаров С. С.** Счётные булевы алгебры и разрешимость. Новосибирск: Науч. книга, 1996. 364 с.

Шиманогов Игорь Николаевич
Вялый Михаил Николаевич

Статья поступила
9 декабря 2024 г.
После доработки —
20 августа 2025 г.
Принята к публикации
22 сентября 2025 г.

INFINITE REGULAR REALIZABILITY PROBLEMS

I. N. Shimanogov^{1, a} and M. N. Vyalyi^{1, 2, 3, b}¹Moscow Institute of Physics and Technology
(National Research University),

1A Bld. 1 Kerchenskaya Street, 117303 Moscow, Russia

²Federal Research Center “Computer Science and Control” RAS
44 Bld. 2 Vavilov Street, 119333 Moscow, Russia³HSE University,

11 Pokrovsky Boulevard, 109028 Moscow, Russia

E-mail: ^ashimanogov.in@phystech.edu, ^bvyalyi@gmail.com

Abstract. A well-studied class of algorithmic problems is that of regular realizability, i.e. checking the non-emptiness of the intersection of a regular language with a given language. This problem has a natural algebraic interpretation which is verifying whether an element of a Boolean algebra belongs to the kernel of a certain homomorphism. This motivates the consideration of a similar problem of infinite regular realizability that is checking whether the intersection of a regular language with a given language is infinite. The paper examines the case of decidable languages and provides a comparative analysis of the complexity of infinite regular realizability problems versus regular realizability problems. Bibliogr. 22.

Keywords: regular language, regular realizability problem, arithmetic hierarchy.

References

1. **A. Bouajjani, J. Esparza, and O. Maler**, Reachability analysis of push-down automata: Application to model-checking, in *CONCUR'97: Concurrency Theory*, Proc. 8th Int. Conf. (Warsaw, Poland, July 1–4, 1997) (Springer, Heidelberg, 1997), pp. 135–150 (Lect. Notes Comput. Sci., V. 1243), DOI: 10.1007/3-540-63141-0_10.

2. **A. Rubtsov** and **M. Vyalyi**, Regular realizability problems and context-free languages, in *Descriptional Complexity of Formal Systems*, Proc. 17th Int. Workshop DCFS 2015 (Waterloo, ON, Canada, June 25–27, 2015) (Springer, Cham, 2015), pp. 256–267 (Lect. Notes Comput. Sci., V. 9118), DOI: 10.1007/978-3-319-19225-3_22.
3. **D. Chistikov**, **R. Majumdar**, and **P. Schepper**, Subcubic certificates for CFL reachability, *Proc. ACM Program. Lang.* **6** (POPL), ID 41 (2022), DOI: 10.1145/3498702.
4. **A. Pavlogiannis**, CFL/Dyck reachability: An algorithmic perspective, *ACM SIGLOG News* **9** (4), 5–25 (2023), DOI: 10.1145/3583660.3583664.
5. **A. H. Kjelstrøm** and **A. Pavlogiannis**, The decidability and complexity of interleaved bidirected Dyck reachability, *Proc. ACM Program. Lang.* **6** (POPL), ID 12 (2022), DOI: 10.1145/3498673.
6. **P. Koutris** and **S. Deep**, The fine-grained complexity of CFL reachability, *Proc. ACM Program. Lang.* **7** (POPL), ID 59 (2023), DOI: 10.1145/3571252.
7. **T. Anderson**, **J. Loftus**, **N. Rampersad**, **N. Santean**, and **J. Shallit**, Detecting palindromes, patterns and borders in regular languages, *Inf. Comput.* **207**, 1096–1118 (2009), DOI: 10.1016/j.ic.2008.06.007.
8. **P. Wolf** and **H. Fernau**, Regular intersection emptiness of graph problems: Finding a needle in a haystack of graphs with the help of automata (Ithaca, NY, 2020) (e-Print Archive / Cornell Univ., arXiv:2003.05826), DOI: 10.48550/arXiv.2003.05826.
9. **P. Wolf**, From decidability to undecidability by considering regular sets of instances, *Theor. Comput. Sci.* **899**, 25–38 (2022), DOI: 10.1016/j.tcs.2021.11.006.
10. **V. Diekert**, **H. Fernau**, and **P. Wolf**, Properties of graphs specified by a regular language, *Acta Inform.* **59**, 357–385 (2022), DOI: 10.1007/s00236-022-00427-z.
11. **P. Wolf**, On the decidability of finding a positive ILP-instance in a regular set of ILP-instances, *Acta Inform.* **59**, 505–519 (2022), DOI: 10.1007/s00236-022-00429-x.
12. **T. Feder** and **M. Y. Vardi**, The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory, *SIAM J. Comput.* **28** (1), 57–104 (1999).
13. **A. A. Bulatov**, A dichotomy theorem for nonuniform CSPs, in *Proc. 58th IEEE Annu. Symp. Foundations of Computer Science* (Berkeley, CA, USA, Oct. 15–17, 2017) (IEEE, Piscataway, 2017), pp. 319–330, DOI: 10.1109/F0CS.2017.37
14. **D. Zhuk**, A proof of CSP dichotomy conjecture, in *Proc. 58th IEEE Annu. Symp. Foundations of Computer Science* (Berkeley, CA, USA, Oct. 15–17, 2017) (IEEE, Piscataway, 2017), pp. 331–342, DOI: 10.1109/F0CS.2017.38
15. **A. A. Bulatov**, A dichotomy theorem for nonuniform CSPs (Ithaca, NY, 2017) (e-Print Archive / Cornell Univ., arXiv:1703.03021), DOI: 10.48550/arXiv.1703.03021.

16. **D. Zhuk**, A proof of the CSP dichotomy conjecture, *J. ACM* **67** (5), ID 30 (2020), DOI: 10.1145/3402029.
17. **M. N. Vyalyi** and **A. A. Rubtsov**, Regular realizability problems for descriptions of finite relations, *Probl. Peredachi Inf.* **60** (3), 46–58 (2024) [Russian] [On universality of regular realizability problems, *Probl. Inf. Transm.* **60** (3), 209–232 (2024), DOI: 10.1134/S0032946024030050].
18. **R. Soare**, *Turing Computability: Theory and Applications* (Springer, Heidelberg, 2016), DOI: 10.1007/978-3-642-3.
19. **D. Kozen**, *Automata and Computability* (Springer, New York, 2012), DOI: 10.1007/978-1-4612-1844-9.
20. **J. O. Shallit**, *A Second Course in Formal Languages and Automata Theory* (Camb. Univ. Press, New York, 2008), DOI: 10.1017/CB09780511808876.
21. **I. N. Shimanogov** and **M. N. Vyalyi**, Classification of relative regular algebras, *Tr. MFTI* **16** (4), 128–134 (2024) [Russian].
22. **S. C. Goncharov**, *Countable Boolean Algebras and Decidability* (Nauchn. Kniga, Novosibirsk, 1996) [Russian].

Igor N. Shimanogov
Mikhail N. Vyalyi

Received December 9, 2024

Revised August 20, 2025

Accepted September 22, 2025