ПРОБЛЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

Том 60 **2024** **Вып. 3**

УДК 621.391 : 519.7

 М.Н. Вялый, А.А. Рубцов

ЗАДАЧИ РЕГУЛЯРНОЙ РЕАЛИЗУЕМОСТИ ДЛЯ ОПИСАНИЙ КОНЕЧНЫХ ОТНОШЕНИЙ[1]

Рассмотрены описания конечных отношений на множестве неотрицательных целых чисел в формате, предложенном П. Вольф и Х. Фернау, и связанные с ними задачи регулярной реализуемости. Доказана универсальность этих задач относительно дизъюнктных сводимостей за полиномиальное время для унарных отношений; относительно дизъюнктных сводимостей на полиномиальной памяти для инвариантных бинарных отношений; а также относительно сводимостей по Тьюрингу с NP-оракулом для инвариантных унарных отношений, заданных в унарном алфавите.

Ключевые слова: регулярная реализуемость, сводимость, NP-оракул.

DOI: 10.31857/S0555292324030069, **EDN:** ZXXKCJ

§ 1. Введение

Класс регулярных языков является одним из важнейших объектов изучения в теории формальных языков. Хорошо известно, что основные алгоритмические задачи теории формальных языков – проверка пустоты, конечности, принадлежности слова регулярному языку – разрешимы, и более того, разрешимы за полиномиальное время, если регулярный язык задан описанием конечного автомата, распознающего язык.

Однако регулярные условия, накладываемые на слова некоторого языка, могут оказаться гораздо сложнее. Соответствующая алгоритмическая задача называется *задачей регулярной реализуемости*. Это целый класс задач, параметризованный языками. Зафиксируем некоторый язык F, называемый *фильтром*. Входом задачи регулярной реализуемости является регулярный язык $L(\mathcal{A})$, распознаваемый автоматом $\mathcal{A}$. Необходимо проверить выполнение условия $L(\mathcal{A}) \cap F \neq \varnothing$. Задачу регулярной реализуемости обозначаем $\mathrm{DRR}(F)$ или $\mathrm{NRR}(F)$ в зависимости от вида автомата $\mathcal{A}$: детерминированный или недетерминированный.

На задачи регулярной реализуемости можно смотреть как на задачи достижимости в графе при ограничении на пути: считаем, что ребра графа помечены символами и интересуемся существованием пути, вдоль которого прочитывается слово из некоторого языка. Наиболее изучена такая задача для ограничений в виде КС языка [1–4]. В этом случае задача регулярной реализуемости разрешима за полиномиальное время.

Общая формулировка задачи регулярной реализуемости дана в [5]. Там же приведены примеры фильтров, для которых задачи регулярной реализуемости полны для некоторых типичных классов сложности или для всего класса перечислимых языков.

[1] Работа выполнена при поддержке Российского научного фонда (грант № 20-11-20203).

Более общим вопросом, чем существование алгоритмически трудных задач, является вопрос об "универсальности" задач некоторого класса среди всех возможных задач разрешения. Более точно, *свойство универсальности* класса языков $\mathscr{C}$ относительно сводимостей $(\leqslant_1, \leqslant_2)$ состоит в том, что для любого непустого языка X существует такой язык $L_X \in \mathscr{C}$, что

$$X \leqslant_1 L_X \leqslant_2 X.$$

Получаем целое семейство свойств универсальности, в котором чем слабее сводимости, тем сильнее свойство. С точки зрения теории алгоритмов имеет смысл рассматривать сводимости, которые не сильнее сводимостей по Тьюрингу. Для применений к теории вычислительной сложности требуются более слабые сводимости.

В [6] доказано свойство универсальности класса DRR относительно пары сводимостей $(\leqslant_{\mathrm{m}}^{\log}, \leqslant_{\mathrm{dtt}}^{\mathrm{FNL}})$ (определения см. в п. 2.2 и в работе [6]). В той же работе доказано, что для каждого класса полиномиальной иерархии существует задача DRR, полная в этом классе относительно полиномиальных сводимостей. Заметим также, что результаты [6] легко переносятся на NRR.

Вопрос об универсальности имеет смысл и для задач регулярной реализуемости с ограниченными классами фильтров. Такого рода результаты можно интерпретировать как универсальность подходящих моделей вычисления. В [6] доказана универсальность DRR относительно той же пары сводимостей, что и в общем случае, для префиксно-замкнутых фильтров. Этот результат говорит о представительности модели автоматов с обобщенным недетерминизмом (см. [7]). Этот результат был уточнен в [8], где была доказана универсальность задачи регулярной реализуемости для фильтров, образованных корректными протоколами работы конечных автоматов с дополнительными структурами данных относительно пары сводимостей $(\leqslant_{\mathrm{m}}^{\log}, \leqslant_{\mathrm{T}}^{\mathrm{P}})$, где $\leqslant_{\mathrm{T}}^{\mathrm{P}}$ – полиномиальные сводимости по Тьюрингу. Отсюда следует, что любой достаточно сильный класс сложности можно описать как класс языков, распознаваемых машинами Тьюринга на логарифмической памяти, к которым добавили структуру данных, соответствующую языку протоколов.

В этой статье мы уточняем границу между классами фильтров со свойствами универсальности и без них. На этот раз мотивация выбора класса фильтров другая и не связана с моделями вычислений.

Задачи регулярной реализуемости (под другим названием – regular intersection emptiness problems) изучались в работах [9, 10]. Там изучен такой круг вопросов: зафиксируем некоторое свойство графов и способ описания графов в виде слов конечного алфавита. Интересует существование графа, обладающего данным свойством, одно из описаний которого удовлетворяет некоторому регулярному ограничению. Другими словами, на вход задачи о графах подается не один граф, а множество графов, описания которых удовлетворяют некоторому регулярному условию – входу задачи NRR(F). Интересует существование хотя бы одного графа в этом множестве, обладающего искомым свойством (которое задает фильтр F). При такой постановке крайне существен способ задания графа. В [9] предложен очень интересный способ кодировки, при котором вершины графа – натуральные числа, а ребра – пары чисел, причем числа записываются в унарной системе. Самой важной особенностью этого способа представления является свобода в выборе порядка записи ребер и возможность повторения элементов списка. Для задач регулярной реализуемости это означает отсутствие "лишних" трудностей, которые могут возникнуть при фиксированном порядке записи ребер (например, см. [8], где порядок записи блоков в протоколе существен).

Для такого способа представления графов в [9] доказана разрешимость задач регулярной реализуемости для многих типичных для теории графов свойств. Ряд родственных результатов получен в работах [10, 11].

Неформально говоря, результаты из [9–11] указывают на структурную простоту выбранного в этих работах способа описания графов. Поэтому неочевиден ответ на вопрос об универсальности задач регулярной реализуемости для таких описаний: раз уж свойства графов легко поддаются алгоритмическому анализу при выбранном способе описания, то строить трудные примеры должно быть сложнее.

Тем не менее в данной статье получены результаты об универсальности задач регулярной реализуемости для описаний графов и унарных отношений на множестве натуральных чисел. Эти результаты используют более сильные сводимости, чем общая теорема об универсальности, что отражает специфику данного класса задач регулярной реализуемости.

При получении таких результатов одним из основных препятствий является свобода в выборе порядка записи элементов множества. Для преодоления этой трудности приходится использовать довольно сложные и нетипичные для теории формальных языков конструкции – эффективные асимптотически хорошие коды. Мы также вводим некоторую новую модель вычисления булевых функций (или, что то же самое, семейств подмножеств некоторого фиксированного множества).

Статья организована следующим образом. В § 2 приводятся точные формулировки неформально поставленных выше задач и напоминаются основные определения, относящиеся к автоматам и алгоритмическим сводимостям. В § 3 описано основное техническое средство анализа задач регулярной реализуемости для таких способов описания: переход к автоматам в алфавите $\mathbb{N}^k$, где k – арность рассматриваемых отношений[2]. Задание таких автоматов опирается на структурную простоту регулярных языков в унарном алфавите.

В § 4 описан наш подход к получению результатов универсальности для различных классов фильтров. Для применения этого подхода к интересующему нас случаю описаний конечных отношений требуется, как уже говорилось, анализ общей задачи представления семейств подмножеств заданного множества (“универсума”) ориентированными графами. Эта задача формулируется в § 5. Именно при построении разделенных семейств множеств возникает необходимость в эффективных асимптотически хороших кодах.

Поскольку основная мотивация статьи связана с задачами регулярной реализуемости для теоретико-графовых свойств, мы рассматриваем отдельно классы отношений на $\mathbb{N}$, инвариантные относительно биекций $\mathbb{N}$. Ведь обычно под свойством графов понимают свойство, которое сохраняется при изоморфизме. Доказательство универсальности для инвариантных свойств удается получить только для весьма сильных сводимостей (дизъюнктные сводимости на полиномиальной памяти).

В § 6 содержатся основные результаты статьи: универсальность описаний унарных отношений (т.е. конечных подмножеств $\mathbb{N}$), универсальность инвариантных бинарных отношений и универсальность инвариантных унарных отношений. Последний результат доказан только для языков в унарном алфавите, поскольку единственный инвариант конечного подмножества – это его размер, что не позволяет строить короткие кодировки двоичных слов числами в унарной системе.

В заключительном параграфе обсуждаются результаты этой статьи и перспективы дальнейших исследований в этом направлении.

Расширенный вариант этой статьи см. в [12].

§ 2. Определения и основные конструкции

Напомним основные определения, связанные с автоматами и трансдьюсерами, сводимостями и классами сложности, а также зафиксируем обозначения. В этом

[2] Под натуральными числами мы понимаем целые неотрицательные числа.

параграфе также фиксируется точный формат описания конечных отношений, используемый в настоящей статье.

2.1. Автоматы и трансдьюсеры. (Недетерминированный) автомат $\mathcal{A}$ в алфавите Σ задается конечным множеством состояний Q, начальным состоянием q_0, множеством принимающих состояний Q_a и отношением переходов

$$\delta_{\mathcal{A}} \subseteq Q \times (\{\varepsilon\} \cup \Sigma) \times Q,$$

где ε – пустое слово. Отношение переходов продолжается до отношения на $Q \times \Sigma^* \times Q$ естественным правилом: $(q_1, w\sigma, q_2) \in \delta_{\mathcal{A}}$ равносильно тому, что существует такое q_3, что $(q_1, w, q_3) \in \delta_{\mathcal{A}}$ и $(q_3, \sigma, q_2) \in \delta_{\mathcal{A}}$. Здесь $w \in \Sigma^*$, $\sigma \in \{\varepsilon\} \cup \Sigma$.

Обратите внимание, что мы допускаем бесконечные алфавиты для автоматов. Это нестандартное определение оказывается здесь удобным. Если алфавит конечный, то возможно задание автомата списком элементов отношения переходов. В случае бесконечного алфавита задание отношения переходов оговаривается особо (см. пример в § 3 ниже).

Слово w *принимается* автоматом $\mathcal{A}$, если $(q_0, w, q_a) \in \delta_{\mathcal{A}}$ для $q_a \in Q_a$. *Языком* называется любое подмножество слов в конечном алфавите. Язык $L(\mathcal{A})$ слов, распознаваемый автоматом $\mathcal{A}$, состоит из слов, принимаемых $\mathcal{A}$. Автоматы $\mathcal{A}'$ и $\mathcal{A}''$ называются *эквивалентными*, если $L(\mathcal{A}') = L(\mathcal{A}'')$.

Переходы (q_1, ε, q_2) называются ε-переходами. По описанию автомата $\mathcal{A}$ возможно построить описание эквивалентного автомата $\mathcal{A}'$ без ε-переходов за полиномиальное время. Поэтому в дальнейшем по умолчанию предполагается, что в автоматах нет ε-переходов (при необходимости применяется описанное выше преобразование).

Ходом автомата $\mathcal{A}$ при чтении слова $w = w_1 w_2 \dots w_n$ называется такая последовательность состояний $q_0, q_1, \dots, q_n$, что $(q_{i-1}, w_i, q_i) \in \delta_{\mathcal{A}}$ для всех $1 \leqslant i \leqslant n$. Ход автомата из начального состояния в принимающее называется *принимающим*.

Всюду далее по умолчанию предполагается, что в автоматах нет *избыточных* состояний, т.е. каждое состояние лежит хотя бы на одном принимающем ходе автомата. Удаление избыточных состояний возможно за полиномиальное от размера описания автомата время.

Трансдьюсер – это конечный автомат, дополненный лентой результата, и для каждого перехода указано, какое слово в *выходном алфавите* трансдьюсер должен дописать на ленту результата к уже записанным там символам. Трансдьюсер T задает бинарное отношение на словах входного и выходного алфавитов: uTv истинно, если при чтении слова u возможен результат v на некотором принимающем ходе. Отношение *рационального доминирования* $A \leqslant_{\mathrm{rat}} B$ на языках означает, что существует такой трансдьюсер T, что

$$A = T(B) = \{v \mid \exists u:\ uTv,\ u \in B\}.$$

2.2. Сводимости. Язык A m-сводится к языку B относительно класса функций $\mathscr{C}$ (обозначение $\leqslant_{\mathrm{m}}^{\mathscr{C}}$), если существует такая функция $f \in \mathscr{C}$, что $x \in A$ равносильно $f(x) \in B$. Чтобы отношение сводимости было предпорядком (транзитивным и рефлексивным отношением), необходимо и достаточно, чтобы класс $\mathscr{C}$ содержал тождественное отображение и был замкнут относительно композиции. Используем обозначения $\leqslant_{\mathrm{m}}^{\mathrm{P}}$ для сводимости за полиномиальное время (сводимость по Карпу) и $\leqslant_{\mathrm{m}}^{\log}$ для сводимости на логарифмической памяти.

Аналогично определяется сводимость по Тьюрингу (сводимость с оракулом): $A \leqslant_{\mathrm{T}}^{\mathscr{C}} B$, если A разрешается алгоритмом из класса $\mathscr{C}$ с оракулом B. (Алгоритм разрешает A, если он вычисляет характеристическую функцию A). В этой статье используется сводимость $\leqslant_{\mathrm{T}}^{\mathrm{FNP}}$, где функции из класса FNP вычислимы за полиномиальное от длины входа время с NP-оракулом. Алгоритм вычисления такой функ-

ции может за один такт работы выяснять у оракула принадлежность слова-запроса к любому языку из NP. Нетрудно видеть, что равносильное требование состоит в том, чтобы алгоритм проверял принадлежность некоторому NP-полному языку.

Пусть R – бинарное отношение из класса P, и пусть для некоторого многочлена $q(\cdot)$ выполняется $|y| \leqslant q(|x|)$ для всех $(x, y) \in R$. Именно такие отношения используются в определении класса NP, и y называется *сертификатом* для x, если $(x, y) \in R$. В построении сводимостей мы используем хорошо известное соображение: алгоритм с NP-оракулом может за полиномиальное время не только проверить существование сертификата, но и найти его.

Частным случаем сводимости по Тьюрингу является табличная сводимость. В этом случае алгоритм разрешения строит список запросов $q_1, \ldots, q_s$ к оракулу и описание булевой функции от s аргументов, которая вычисляет результат по ответам оракула на запросы. Нам потребуется еще более ограниченный тип табличных сводимостей: *дизъюнктная сводимость* (dtt), когда вычисляется дизъюнкция ответов на запросы (т.е. ответ положительный, если хотя бы одно слово-запрос принадлежит языку-оракулу).

Мы будем использовать два вида дизъюнктных сводимостей. Более сильная из них, $\leqslant_{\mathrm{dtt}}^{\mathrm{PSPACE}}$, – дизъюнктная сводимость на полиномиальной памяти. Заметим, что в этом случае список запросов может быть экспоненциально велик по сравнению с длиной входа. Такая сводимость имеет смысл для достаточно трудных языков. Например, соответствующий ей предпорядок различает классы арифметической иерархии.

Более слабая, $\leqslant_{\mathrm{dtt}}^{\mathrm{P}}$, использует функции, вычислимые за полиномиальное время.

2.3. Описания конечных отношений. Нас интересуют фильтры, которые задают конечные отношения (например, графы). Обобщая кодировки из [8, 9], будем рассматривать фильтры следующего вида. Пусть $R \subseteq \mathbb{N}^k$ – конечное k-арное отношение. Кодируем последовательность $x = (x_1, x_2, \ldots, x_k) \in \mathbb{N}^k$ словом

$$\langle x \rangle = \lhd a^{x_1} \# a^{x_2} \# \ldots \# a^{x_k} \rhd$$

в алфавите $\{a, \lhd, \rhd, \#\}$. Будем называть такие слова *блоками*. *Описанием* $\langle R \rangle$ конечного отношения R является любое слово вида

$$\prod_{i=1}^{t} \langle x(i) \rangle, \quad \text{где} \quad \{x(i) : 1 \leqslant i \leqslant t\} = R.$$

Здесь $\prod$ обозначает конкатенацию слов. Заметим, что порядок записи кодировок различных $x \in R$ произвольный и допустимы повторения. Таким образом, кодировка неоднозначна.

Сопоставим семейству конечных отношений $\mathcal{R}$ язык

$$\langle \mathcal{R} \rangle = \{\langle R \rangle : R \in \mathcal{R}\},$$

состоящий из всех описаний всех отношений из $\mathcal{R}$. Эти языки будут фильтрами в задачах регулярной реализуемости, которые рассматриваются в данной статье.

Как и в [9], нас будут также интересовать описания простых неориентированных графов без изолированных вершин. Следуя [9], полагаем, что ребра графа перечисляются в произвольном порядке, порядок вершин в описании ребра не важен, диагональные блоки $\lhd a^x \# a^x \rhd$ допустимы, но игнорируются. Более точно, произвольному бинарному отношению R сопоставляем граф $G_R = (V, E)$ по правилу

$$\begin{aligned} V &= \{v : \exists u\ (u, v) \in R\} \cup \{u : \exists v\ (u, v) \in R\}, \\ E &= \big\{\{u, v\} : (u \neq v) \wedge \big(((u, v) \in R) \vee ((v, u) \in R)\big)\big\}. \end{aligned} \tag{1}$$

По определению описанием $\langle G\rangle$ графа G является описание $\langle R\rangle$ любого отношения R, для которого $G = G_R$.

Наши кодировки несколько отличаются от кодировок в других работах на эту тему, см. [9–11]. Различие состоит в формате разделителей между описаниями чисел в унарной системе. Поэтому одни кодировки переводятся в другие с помощью трансдьюсерного преобразования, и наоборот. Нас интересуют задачи регулярной реализуемости, их сложность не изменяется при таких преобразованиях, так как в [3] доказано, что из $A \leqslant_{\mathrm{rat}} B$ следует $\mathrm{NRR}(A) \leqslant_{\mathrm{m}}^{\log} \mathrm{NRR}(B)$. Поэтому эти различия в кодировках несущественны.

Обычно интересуются свойствами графов, сохраняющимися при изоморфизме. Будем далее называть *инвариантными* семейства отношений на $\mathbb{N}$, замкнутые относительно биекций $\mathbb{N}$. Из определения (1) сразу следует, что описания семейства графов с вершинами из $\mathbb{N}$, замкнутого относительно изоморфизма, образуют инвариантное семейство отношений.

§ 3. Автоматы в бесконечном алфавите

Рассмотрим автомат $\mathcal{A}$ в алфавите $\{a, \lhd, \rhd, \#\}$. Обозначим через $\mathcal{R}^k(\mathcal{A})$ класс тех k-арных отношений на $\mathbb{N}$, описания которых принимаются автоматом $\mathcal{A}$: хотя бы одно описание для каждого отношения. Слова в унарном алфавите находятся во взаимно-однозначном соответствии с натуральными числами, слову сопоставляется его длина. В дальнейшем анализе удобно избавиться от символов-разделителей и рассматривать автоматы в бесконечном алфавите $\mathbb{N}^k$.

Для этого по автомату $\mathcal{A}$ построим автомат $\widetilde{\mathcal{A}}^k_{\mathbb{N}}$ с тем же множеством состояний, теми же начальными и принимающими состояниями и алфавитом $\mathbb{N}^k$. Отношение переходов определим так: для всех $\boldsymbol{i} = (i_1, i_2, \ldots, i_k) \in \mathbb{N}^k$

$$(q_1, \boldsymbol{i}, q_2) \in \delta_{\widetilde{\mathcal{A}}^k_{\mathbb{N}}} \iff (q_1, \lhd a^{i_1} \# a^{i_2} \# \ldots \# a^{i_k} \rhd, q_2) \in \delta_{\mathcal{A}}\,. \tag{2}$$

Из такого определения следует, что если непустое слово в алфавите $\mathbb{N}^k$ принадлежит $L(\widetilde{\mathcal{A}}^k_{\mathbb{N}})$, то множество входящих в него символов (т.е. последовательностей неотрицательных целых чисел длины k) принадлежит $\mathcal{R}^k(\mathcal{A})$. Верно и обратное: любое отношение $R \in \mathcal{R}^k(\mathcal{A})$ является множеством символов некоторого слова из $L(\widetilde{\mathcal{A}}^k_{\mathbb{N}})$. В автомате $\widetilde{\mathcal{A}}^k_{\mathbb{N}}$ возможны избыточные состояния. После их удаления получаем автомат $\mathcal{A}^k_{\mathbb{N}}$, для которого выполняются те же соотношения с множеством $\mathcal{R}^k(\mathcal{A})$. Об отношениях из $\mathcal{R}^k(\mathcal{A})$ будем говорить, что они принимаются автоматом $\mathcal{A}^k_{\mathbb{N}}$. Аналогично, граф G принимается автоматом, если для некоторого отношения R выполняется $G = G_R$ и $R \in L(\mathcal{A}^2_{\mathbb{N}})$.

Представление автомата в бесконечном алфавите в виде конечного объекта требует уточнения в описании отношения переходов. В данном случае это довольно просто сделать в силу ограниченной структуры регулярных языков в унарном алфавите. Теорема Хробака–Мартинеза (см. [13,14] и уточнения в [15,16]) утверждает, что по автомату в унарном алфавите, имеющему n состояний, можно за полиномиальное время построить описание множества длин принимаемых ими слов в виде $O(n^2)$ арифметических прогрессий $\{a + bt : t \in \mathbb{N}\}$, где $a = O(n^2)$, $b = O(n)$. Если не оговорено противное, далее мы предполагаем, что одномерные полулинейные множества представлены в виде объединения арифметических прогрессий с указанными ограничениями на начальные члены и разности.

Сопоставим паре состояний (q_1, q_2) автомата $\mathcal{A}^k_{\mathbb{N}}$ множество $L_{q_1 q_2} \subseteq \mathbb{N}^k$ тех наборов из $\mathbb{N}^k$, чтение которых переводит q_1 в q_2.

Лемма 1. *Множество* $L_{q_1q_2} \subseteq \mathbb{N}^k$, *где* $k = O(1)$, *является объединением декартовых произведений одномерных полулинейных множеств. Размер описания этого объединения* $\mathrm{poly}(\mathrm{size}(\mathcal{A}))$, *где* $\mathrm{size}(\mathcal{A})$ – *размер описания* $\mathcal{A}$.

Отношение переходов автомата $\mathcal{A}^k_{\mathbb{N}}$, т.е. множество $\{(q_1, \boldsymbol{i}, q_2) : \boldsymbol{i} \in L_{q_1q_2}\}$, задаем соответствующими $L_{q_1q_2}$ объединениями декартовых произведений одномерных полулинейных множеств. Размер такого описания полиномиально ограничен размером описания исходного автомата $\mathcal{A}$, как следует из леммы 1. Построение описания автомата $\mathcal{A}^k_{\mathbb{N}}$ по $\mathcal{A}$ возможно за полиномиальное время в силу теоремы Хробака – Мартинеса.

В дальнейшем будут важны те автоматы, для которых $\mathcal{R}^k(\mathcal{A})$ конечно. Поэтому потребуется следующий факт.

Лемма 2. $\mathcal{R}^k(\mathcal{A})$ *конечно тогда и только тогда, когда* $L_{q_1q_2}$ *конечно для любой пары* (q_1, q_2). *Аналогичное утверждение выполняется также и для множества принимаемых автоматом* $\mathcal{A}$ *графов.*

Следствие 1. *Конечность* $\mathcal{R}^k(\mathcal{A})$ *проверяется за полиномиальное время.*

§ 4. Общая схема доказательства универсальности

Универсальность описаний отношений заданного типа состоит в том, что для любого непустого языка двоичных слов X существует такой класс $\mathcal{U}_X$ конечных k-арных отношений этого типа, что

$$X \leqslant_1 \mathrm{NRR}\big(\langle \mathcal{U}_X \rangle\big) \leqslant_2 X. \tag{3}$$

Точный вид сводимостей и требования к классу отношений будут в разных случаях различными. Однако общая схема доказательства одна и та же, и мы ее сейчас опишем.

Определяем функцию $X \mapsto \mathcal{U}_X$ следующим образом. Обозначим через $P_f(\mathbb{N}^k)$ семейство всех конечных подмножеств $\mathbb{N}^k$, т.е. конечных k-арных отношений на $\mathbb{N}$. Выберем кодировку $\varphi\colon \{0,1\}^* \to P_f(\mathbb{N}^k)$ двоичных слов конечными отношениями, удовлетворяющую следующим свойствам:

1. φ инъективна;
2. φ вычислима за полиномиальное время;
3. (частичная) функция φ^{-1} также вычислима за полиномиальное время;
4. $x \in X$ равносильно $\varphi(x) \in \mathcal{U}_X$;
5 (условие конечности). Для всякого автомата $\mathcal{A}$ из бесконечности $\mathcal{R}^k(\mathcal{A})$ следует, что $\mathcal{A}$ принимает хотя бы одно описание $\langle R \rangle$ для некоторого $R \notin \varphi(\{0,1\}^*)$.

Для выполнения условия 4 потребуем

$$\mathcal{U}_X = \varphi(X) \cup \overline{\varphi(\{0,1\}^*)} \tag{4}$$

Тогда

$$\mathcal{U}_X \cap \varphi\big(\{0,1\}^*\big) = \varphi(X),$$

и сводимость $X \leqslant \mathrm{NRR}\big(\langle \mathcal{U}_X \rangle\big)$ возможно определить как $x \mapsto \mathcal{A}_{\langle \varphi(x) \rangle}$, где автомат $\mathcal{A}_{\langle \varphi(x) \rangle}$ принимает в точности одно слово из множества описаний отношения $\varphi(x)$ (такие сводимости в [6] названы *моносводимостями*). Из условия 2 следует, что в таком случае

$$X \leqslant^{\mathrm{P}}_{\mathrm{m}} \mathrm{NRR}\big(\langle \mathcal{U}_X \rangle\big).$$

Сводимость $\mathrm{NRR}\big(\langle \mathcal{U}_X \rangle\big) \leqslant X$ определяется по-разному на двух классах входов. Мы разделяем автоматы, т.е. входы $\mathrm{NRR}\big(\langle \mathcal{U}_X \rangle\big)$, на *тривиальные* и *нетривиальные.*

Определения тривиальности варьируются. Однако всегда для тривиального автомата $\mathcal{A}$ выполняется

$$\mathcal{R}^k(\mathcal{A}) \cap \overline{\varphi(\{0,1\}^*)} \neq \varnothing,$$

т.е. $\mathcal{A}$ принимает хотя бы одно слово $\langle R \rangle$ для отношения R, которое не является кодировкой двоичного слова. Важный частный случай этого условия – бесконечность $\mathcal{R}^k(\mathcal{A})$, так как тогда из условия конечности 5 следует

$$\mathcal{R}^k(\mathcal{A}) \cap \overline{\varphi(\{0,1\}^*)} \neq \varnothing.$$

В силу (4) ответ в задаче $\mathrm{NRR}\big(\langle \mathcal{U}_X \rangle\big)$ для тривиального автомата $\mathcal{A}$ положительный, т.е. $\mathcal{A} \in \mathrm{NRR}\big(\langle \mathcal{U}_X \rangle\big)$. Это позволяет определять вторую сводимость

$$\mathrm{NRR}\big(\langle \mathcal{U}_X \rangle\big) \leqslant X$$

на тривиальных автоматах простейшим способом: отображать каждый тривиальный автомат в некоторый фиксированный элемент $x_0 \in X$. Такое определение сводимости возможно для тех сводимостей, сложность которых достаточна для проверки тривиальности автомата. Формально, дизъюнктная сводимость на тривиальном автомате выдает список запросов, состоящий из единственного элемента x_0.

Для нетривиальных автоматов возможны оба ответа в задаче $\mathrm{NRR}\big(\langle \mathcal{U}_X \rangle\big)$. Пусть $\mathcal{A}$ – нетривиальный автомат, в частности, $|\mathcal{R}^k(\mathcal{A})| < \infty$. Список запросов к оракулу X состоит из слов

$$\varphi^{-1}(\mathcal{R}^k(\mathcal{A})) = \{\varphi^{-1}(R) : \ R \in \mathcal{R}^k(\mathcal{A})\},$$

здесь φ – та же функция, что и при построении первой сводимости. Поэтому

$$\mathcal{A} \in \mathrm{NRR}\big(\langle \mathcal{U}_X \rangle\big) \iff \varphi^{-1}(\mathcal{R}^k(\mathcal{A})) \cap X \neq \varnothing,$$

поскольку хотя бы один запрос возвращает положительный ответ в точности при $\mathcal{A} \in \mathrm{NRR}\big(\langle \mathcal{U}_X \rangle\big)$. Количество запросов конечно, так как φ – инъекция по условию 1 и $|\mathcal{R}^k(\mathcal{A})| < \infty$.

Так как множество $\mathcal{R}^k(\mathcal{A})$ конечно, автомат $\mathcal{A}^k_{\mathbb{N}}$ принимает только отношения размера $\mathrm{poly}(\mathrm{size}(\mathcal{A}))$ в силу лемм 1 и 2. Все эти отношения можно перечислить на полиномиальной памяти. Это уже дает сводимость на полиномиальной памяти. В некоторых случаях она может быть ослаблена до дизъюнктной полиномиальной $\leqslant^{\mathrm{P}}_{\mathrm{dtt}}$.

В случае инвариантных отношений эту схему нужно модифицировать. В этом случае нужны классы изоморфизма конечных отношений, поэтому $\mathcal{U}_X$ должно быть замкнуто относительно биекций $\mathbb{N}$. Значит, нетривиальны те автоматы $\mathcal{A}$, для которых $\mathcal{R}^k(\mathcal{A})$ конечно и содержит лишь отношения, изоморфные отношениям из $\varphi(\{0,1\}^*)$. Для второй сводимости нужен список всех классов изоморфизма отношений из $\mathcal{R}^k(\mathcal{A})$.

Заметим также, что для универсальности инвариантных унарных отношений (теорема 3 ниже) условие конечности 5 требуется ослабить.

§ 5. Пути на размеченных графах

Для реализации изложенного выше плана важно находить все отношения из $\mathcal{R}^k(\mathcal{A})$, если это множество конечно. По лемме 2 равносильное условие состоит в том, что множества $L_{q_1 q_2}$ конечные для всех пар q_1, q_2 состояний автомата $\mathcal{A}^k_{\mathbb{N}}$ (их может быть меньше, чем состояний $\mathcal{A}$, см. § 3).

Поскольку порядок перечисления элементов в описании отношения несущественен, характеризация принимаемых автоматом $\mathcal{A}^k_{\mathbb{N}}$ отношений дается следующей общей моделью задания семейств конечных множеств ориентированными графами.

Рассматриваем ориентированные графы с петлями и кратными ребрами. *Граф переходов* $G = (V, s, t, E, \ell)$ задается множеством вершин V, множеством ориентированных ребер E, начальной вершиной s, конечной вершиной t и функцией разметки

$$\ell\colon E \to \{\varnothing\} \cup U,$$

которая отображает ребра графа переходов в пустое множество или в элемент конечного множества U, именуемого далее *универсумом меток*. Размер $|G|$ графа переходов – это размер описания G списками вершин, ребер и таблицей функции разметки.

Значение функции $\ell(e)$, $e \in E$, называем *меткой* ребра e. Для пути P *множеством меток пути* $\ell(P)$ называем множество всех непустых меток его ребер. Говорим также, что P *несет* множество меток $\ell(P)$. Через $\mathcal{F}(G)$ обозначаем семейство множеств меток путей из s в t (в дальнейшем (s,t)-пути).

Связь с автоматами $\mathcal{A}^k_{\mathbb{N}}$, у которых все множества $L_{q_1q_2}$ конечные, прямолинейна. Вершинами графа G являются состояния автомата и дополнительная терминальная вершина, куда ведут ребра из принимающих состояний, помеченные пустым множеством. Каждому $\boldsymbol{i} \in L_{q_1q_2}$ соответствует ребро (q_1, q_2) в графе переходов G (разным $\boldsymbol{i}$ соответствуют разные ребра). Меткой этого ребра является $\boldsymbol{i}$. Начальной вершиной является начальное состояние $\mathcal{A}^k_{\mathbb{N}}$. Так как $\mathcal{R}^k(\mathcal{A})$ совпадает с семейством множеств символов непустых слов языка $L(\mathcal{A}^k_{\mathbb{N}})$, то

$$\mathcal{R}^k(\mathcal{A}) = \mathcal{F}(G).$$

Проверка принадлежности множества S семейству $\mathcal{F}(G)$ алгоритмически трудна.

Задача ALL-LABELS. Вход: граф переходов $G = (V, s, t, E, \ell)$. Требуется выяснить, существует ли такой (s,t)-путь P в G, что $\ell(P) = \ell(E)$.

Лемма 3. *Задача* ALL-LABELS *является* NP*-полной.*

Следствие 2. *Проверка принадлежности множества S семейству $\mathcal{F}(G)$ является* NP*-полной.*

Хотя проверка $S \in \mathcal{F}(G)$ является NP-полной, множества меток путей возможно эффективно порождать в смысле инкрементального порождения с полиномиальной задержкой, когда каждый следующий элемент множества порождается за время, полиномиальное от длины входа и количества уже найденных элементов множества (см., например, [17]). Более точно, для инкрементального порождения $\mathcal{F}(G)$ требуется решение следующей задачи.

Задача NEXT-LABEL. Вход: граф переходов $G = (V, s, t, E, \ell)$ и семейство множеств меток

$$\mathcal{S} = \{S_0, \ldots, S_{t-1}\}, \quad \mathcal{S} \subseteq \mathcal{F}(G).$$

Требуется проверить равенство $\mathcal{F}(G) = \mathcal{S}$, и в случае отрицательного ответа предъявить множество $S_t \in \mathcal{F}(G) \setminus \mathcal{S}$.

Для ациклических графов переходов задача NEXT-LABEL разрешима за полиномиальное от длины входа время. Поэтому эффективное порождение $\mathcal{F}(G)$ возможно для ациклического графа переходов, если $|\mathcal{F}(G)|$ полиномиально ограничено $|G|$.

Ограничение ациклическими графами несущественно.

Лемма 4. *По графу переходов G за полиномиальное время можно построить такой ациклический граф переходов G', что $\mathcal{F}(G') = \mathcal{F}(G)$.*

Лемма 5. *Задача* NEXT-LABEL *разрешима за полиномиальное время.*

Общий план построения сводимостей из § 4 требует порождения всех множеств из $\mathcal{F}(G)$. Как следует из леммы 5, это возможно за полиномиальное время, если $|\mathcal{F}(G)|$ полиномиально ограничено относительно $|G|$. Для выполнения этого условия нам потребуются ε-разделенные семейства множеств. Через $S_1 \oplus S_2$ обозначаем симметрическую разность множеств. Семейство конечных подмножеств $\mathcal{F}$ назовем *ε-разделенным*, если для каждой пары различных множеств $S_1 \neq S_2$ из $\mathcal{F}$ выполняется

$$|S_1 \oplus S_2| > \varepsilon \cdot \max\big(|S_1|, |S_2|\big)$$

при

$$\varepsilon \cdot \max\big(|S_1|, |S_2|\big) \geqslant 6.$$

Последнее ограничение носит технический характер, добавлено для упрощения доказательств. Из него сразу следует, что $\varepsilon > 0$. Заметим также, что $\varepsilon < 2$ для любого ε-разделенного семейства. Эта оценка достигается: семейство непересекающихся множеств, мощности которых одинаковы и не меньше 3, является $(2-\delta)$-разделенным при любом δ, таком что $2 > \delta > 0$.

Лемма 6. *Пусть G – такой граф переходов с n вершинами и универсумом размера m, что $\mathcal{F}(G)$ является ε-разделенным для некоторого ε. Тогда*

$$|\mathcal{F}(G)| \leqslant m \cdot n^{3/\varepsilon} + m^{6/\varepsilon}.$$

Сформулируем алгоритмический вариант леммы 6.

Лемма 7. *Пусть $\mathscr{C}$ – такое семейство подмножеств, что $\mathscr{C} \in \mathrm{P}$, и пусть для любого графа переходов G из $\mathcal{F}(G) \subseteq \mathscr{C}$ следует, что $\mathcal{F}(G)$ является ε-разделенным для некоторого $\varepsilon = \varepsilon(G)$, где функция $\varepsilon(G)$ полиномиально вычислима. Тогда существует алгоритм, который по графу переходов G проверяет, что $\mathcal{F}(G) \subseteq \mathscr{C}$, и в случае положительного ответа порождает список множеств из $\mathcal{F}(G)$ за время*

$$\mathrm{poly}(|G|^{O(1/\varepsilon(G))}).$$

Для приложения к доказательству универсальности описаний унарных отношений потребуются *обильные* разделенные семейства множеств, т.е. такие семейства, в которые входит экспоненциально много множеств относительно размера универсума m. Кроме того, для применения леммы 7 необходимо, чтобы такие семейства были эффективно разрешимыми. Для этих целей понадобятся эффективные асимптотически хорошие двоичные коды.

Напомним, что линейным двоичным (n, k, d)-кодом C называется такое подпространство размерности k координатного векторного пространства $\mathbb{F}_2^n$, что среди координат каждого ненулевого вектора этого подпространства не менее d единиц. Асимптотически хорошим называется такой код, для которого

$$k = \Omega(n) \quad \text{и} \quad d = \Omega(n).$$

Эффективность в данном случае означает лишь существование полиномиального алгоритма кодирования: такого семейства линейных отображений

$$c_k \colon \mathbb{F}_2^k \to \mathbb{F}_2^n,$$

что $c_k(\mathbb{F}_2^k) = C$ и функция $f(k, x) = c_k(x)$ вычислима за время $\mathrm{poly}(k, n)$. Более подробные сведения о корректирующих кодах можно найти в [18, 19]. Там же описаны конструкции эффективных асимптотически хороших кодов.

Мы используем эффективные асимптотически хорошие коды для построения кодировки φ, описанной в § 4.

§ 6. Результаты об универсальности

Теорема 1. *Для любого языка $\varnothing \subset X \subseteq \{0,1\}^*$ существует такое семейство $\mathcal{U}_X$ конечных подмножеств $\mathbb{N}$, что*

$$X \leqslant_{\mathrm{m}}^{\mathrm{P}} \mathrm{NRR}\big(\langle \mathcal{U}_X \rangle\big) \leqslant_{\mathrm{dtt}}^{\mathrm{P}} X.$$

Теорема 2. *Для любого языка $\varnothing \subset X \subseteq \{0,1\}^*$ существует такое семейство графов $\mathcal{G}_X$, замкнутое относительно изоморфизма, что*

$$X \leqslant_{\mathrm{m}}^{\mathrm{P}} \mathrm{NRR}\big(\langle \mathcal{G}_X \rangle\big) \leqslant_{\mathrm{dtt}}^{\mathrm{PSPACE}} X.$$

Единственный инвариант конечных подмножеств $\mathbb{N}$ относительно биекций $\mathbb{N}$ – это размер (мощность) множества. Поэтому инвариантное семейство $\mathcal{I}_L \subset 2^{\mathbb{N}}$ конечных унарных отношений задается подмножеством $L \subseteq \mathbb{N}$ и состоит из таких конечных подмножеств натуральных чисел, мощности которых принадлежат L. Это ограничивает возможности кодировки двоичных слов, так как для слов длины n потребуются экспоненциально большие числа, а числа здесь представляются в унарной кодировке. Поэтому в данном случае мы ограничимся более слабым результатом и докажем универсальность инвариантных унарных отношений для языков в унарном алфавите.

Теорема 3. *Для любого языка $\varnothing \subset X \subseteq \{a\}^*$ существует такое множество $L \subset \mathbb{N}$, что*

$$X \leqslant_{\mathrm{m}}^{\mathrm{P}} \mathrm{NRR}\big(\langle \mathcal{I}_L \rangle\big) \leqslant_{\mathrm{T}}^{\mathrm{FNP}} X.$$

§ 7. Заключение

Эта статья расширяет список классов фильтров, для которых выполняется свойство универсальности. Сравнивая известные примеры – произвольные и префиксно-замкнутые фильтры из [6], фильтры, которые являются языками корректных протоколов работы конечных автоматов с дополнительными структурами данных, из [8] – с новыми классами фильтров, можно отметить следующие особенности. Во-первых, усиление ограничений на фильтры приводит к усилению требуемых сводимостей, т.е. дает более слабые свойства универсальности. Это ожидаемое явление. Во-вторых, для доказательства универсальности новых классов фильтров потребовались более сложные технические инструменты. В частности, существенную роль в доказательствах этой статьи играют асимптотически хорошие эффективные коды. Это необычно для теории формальных языков. Кроме того, для доказательств универсальности пришлось изучить представление семейств множеств путями в ориентированных графах. Это новая неоднородная модель вычислений, свойства которой существенно отличаются от большинства стандартных неоднородных моделей вычисления. Мы надеемся, что эта модель найдет и другие приложения в теоретической информатике. В-третьих, во всех известных примерах классы фильтров, несмотря на структурные ограничения, параметризованы произвольными языками или их аналогами (например, семействами конечных отношений в данной статье).

Возможно ли наложить такие сильные структурные ограничения при сохранении параметризации произвольными языками, чтобы полученный класс фильтров не был универсальным даже в самом слабом разумном смысле общих сводимостей по Тьюрингу? Ответ на этот вопрос неизвестен, хотя кажется правдоподобным, что такой пример возможно построить, используя методы теории алгоритмов. Однако при таком построении пример окажется весьма искусственным.

Второй интересный вопрос, возникающий при сравнении полученных результатов, связан со сложностью сводимостей, требуемых для универсальности. Возможно ли доказать универсальность инвариантных свойств графов, ограничиваясь полиномиальными дизъюнктными сводимостями (или хотя бы полиномиальными сводимостями по Тьюрингу)? Ответ на такой вопрос, даже по модулю стандартных теоретико-сложностных гипотез, был бы чрезвычайно интересным. Нам представляется, что для этого необходимы новые технические средства.

СПИСОК ЛИТЕРАТУРЫ

1. *Yannakakis M.* Graph-Theoretic Methods in Database Theory // Proc. 9th ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems (PODS'90). Nashville, TN, USA. Apr. 2–4, 1990. New York: ACM, 1990. P. 230–242. https://doi.org/10.1145/298514.298576
2. *Bouajjani A., Esparza J., Maler O.* Reachability Analysis of Pushdown Automata: Application to Model-Checking // Proc. Int. Conf. on Concurrency Theory (CONCUR'97). Warsaw, Poland. July 1–4, 1997. Lect. Notes Comput. Sci. V. 1243. Berlin, Heidelberg: Springer, 1997. P. 135–150. https://doi.org/10.1007/3-540-63141-0_10
3. *Rubtsov A.A., Vyalyi M.N.* Regular Realizability Problems and Context-Free Languages // Proc. 17th Int. Workshop on Descriptional Complexity of Formal Systems (DCFS'2015). Waterloo, ON, Canada. June 25–27, 2015. Lect. Notes Comput. Sci. V. 9118. Berlin, Heidelberg: Springer, 2015. P. 256–267. https://doi.org/10.1007/978-3-319-19225-3_22
4. *Chistikov D., Majumdar R., Schepper P.* Subcubic Certificates for CFL Reachability // Proc. ACM Program. Lang. 2022. V. 6. Article 41 (29 pp.). https://doi.org/10.1145/3498702
5. *Вялый М.Н.* О задачах регулярной реализуемости // Пробл. передачи информ. 2011. Т. 47. № 4. С. 43–54. https://www.mathnet.ru/rus/ppi2059
6. *Вялый М.Н.* О выразительной силе задач регулярной реализуемости // Пробл. передачи информ. 2013. Т. 49. № 3. С. 86–104. https://www.mathnet.ru/rus/ppi2117
7. *Vyalyi M.N.* On Models of a Nondeterministic Computation // Computer Science – Theory and Applications: Proc. 4th Int. Computer Science Symp. in Russia (CSR'09). Novosibirsk, Russia. Aug. 18–23, 2009. Lect. Notes Comput. Sci. V. 5675. Berlin, Heidelberg: Springer, 2009. P. 334–345. https://doi.org/10.1007/978-3-642-03351-3_31
8. *Rubtsov A., Vyalyi M.* Automata Equipped with Auxiliary Data Structures and Regular Realizability Problems, https://arxiv.org/abs/2210.03934 [cs.FL], 2022.
9. *Wolf P., Fernau H.* Regular Intersection Emptiness of Graph Problems: Finding a Needle in a Haystack of Graphs with the Help of Automata, https://arxiv.org/abs/2003.05826 [cs.FL], 2020.
10. *Wolf P.* From Decidability to Undecidability by Considering Regular Sets of Instances // Theor. Comput. Sci. 2022. V. 899. P. 25–38. https://doi.org/10.1016/j.tcs.2021.11.006
11. *Diekert V., Fernau H., Wolf P.* Properties of Graphs Specified by a Regular Language // Acta Inform. 2022. V. 59. № 4. P. 357–385. https://doi.org/10.1007/s00236-022-00427-z
12. *Rubtsov A., Vyalyi M.* On Universality of Regular Realizability Problems // Probl. Inf. Transm. 2024. V. 60. № 3. P. 209–232. https://doi.org/10.1134/S0032946024030050
13. *Chrobak M.* Finite Automata and Unary Languages // Theor. Comput. Sci. 1986. V. 47. P. 149–158. https://doi.org/10.1016/0304-3975(86)90142-8
14. *Martinez A.* Efficient Computation of Regular Expressions from Unary NFAs // Pre-proc. 4th Int. Workshop on Descriptional Complexity of Formal Systems (DCFS 2002). London, Canada. Aug. 21–24, 2002. Dept. Comput. Sci., Univ. of Western Ontario, Canada, 2002. P. 174–187.
15. *Chrobak M.* Errata to: "Finite Automata and Unary Languages": [Theoret. Comput. Sci. 47 (1986) 149–158] // Theor. Comput. Sci. 2003. V. 302. № 1–3. P. 497–498. https://doi.org/10.1016/S0304-3975(03)00136-1

16. *To A.W.* Unary Finite Automata vs. Arithmetic Progressions // Inform. Process. Lett. 2009. V. 109. № 17. P. 1010–1014. https://doi.org/10.1016/j.ipl.2009.06.005
17. *Gurvich V.* Complexity of Generation // Computer Science – Theory and Applications: Proc. 13th Int. Computer Science Symp. in Russia (CSR 2018). Moscow, Russia. June 6–10, 2018. Lect. Notes Comput. Sci. V. 10846. Berlin, Heidelberg: Springer, 2018. P. 1–14. https://doi.org/10.1007/978-3-319-90530-3_1
18. *Мак-Вильямс Ф.Дж., Слоэн Н.Дж.А.* Теория кодов, исправляющих ошибки. М.: Связь, 1979.
19. *Ромащенко А.Е., Румянцев А.Ю., Шень А.* Заметки по теории кодирования. М.: МЦНМО, 2017.

Вялый Михаил Николаевич
Рубцов Александр Александрович
Национальный исследовательский университет
“Высшая школа экономики”, Москва
vyalyi@gmail.com
rubtsov99@gmail.com

Поступила в редакцию
05.09.2024
После доработки
17.10.2024
Принята к публикации
22.10.2024