

Полурешётки подмножеств потенциальных корней в задачах теории формальных языков.

Часть III. Условие существования решётки

Б. Ф. Мельников

Аннотация—В статье мы рассматриваем все возможные подмножества множества потенциальных корней, образующие в некоторых ситуациях полурешётки – по пересечению и/или по объединению. Такие структуры возникают в двух похожих задачах теории формальных языков. Конкретно, для некоторого заданного конечного языка мы рассматриваем задачу извлечения корня заданной степени, а также задачу построения оптимального инверсного морфизма – где оптимальность можно определить, например, как длину максимального слова языка, являющегося инверсным морфическим образом.

В обоих приведённых случаях необходимо построить множество так называемых потенциальных корней – т.е. таких слов рассматриваемого алфавита, для каждого из которых некоторая его степень входит в исходный язык. При этом важно отметить, что построение множества всех потенциальных корней может быть выполнено с помощью несложного полиномиального алгоритма. Экспоненциальные алгоритмы для обеих задач очевидны: надо просто перебрать все подмножества множества этих потенциальных корней, и среди этих подмножеств выбрать подходящее. Поэтому проблема состоит в описании возможных полиномиальных алгоритмов для этих задач.

Для обеих указанных задач вызывает интерес возможное существование двух полурешёток, имеющих на предварительно построенном множестве подмножеств потенциальных корней.

Среди прочего, мы в статье приводим формулировку одной важной гипотезы теории формальных языков – при выполнении которой мы можем утверждать, что специальное подмножество множества языков, каждый из которых является инверсным морфическим образом заданного языка, образует не только полурешётку по объединению, но и полурешётку по пересечению (а, следовательно, решётку).

Предлагаемая третья часть статьи, аналогично второй части, в большей степени посвящена второй из рассматриваемых задач – задаче построения оптимального инверсного морфизма. Мы формулируем гипотезу теории формальных языков, при выполнении которой специальные подмножества множества потенциальных корней образуют полурешётку по пересечению.

Ключевые слова—формальные языки, итерации языков, морфизмы, инверсные морфизмы, бинарные отношения, полурешётки, алгоритмы.

Предлагаемая часть III статьи является продолжением [1], [2]. Здесь мы продолжаем нумерацию разделов и формул, а нумерация ссылок на использованную литературу – новая.

Статья получена 17 марта 2022 г.

Борис Феликсович Мельников, Университет МГУ–ППИ в Шэньчжэне (bf-melnikov@yandex.ru).

ХIII. ОБ ОДНОЙ ГИПОТЕЗЕ ТЕОРИИ ФОРМАЛЬНЫХ ЯЗЫКОВ – РЕЗУЛЬТАТЫ ПРЕДЫДУЩИХ ПУБЛИКАЦИЙ

В этом и двух следующих разделах рассматривается гипотеза теории формальных языков, которую мы считаем очень важной. Очень кратко эту гипотезу можно сформулировать следующим образом: очевидное *достаточное* условие выполнения отношения \Leftrightarrow является *необходимым и достаточным*.

Есть «ещё одна преамбула», для её формулировки повторим мысль из раздела IV (часть I) настоящей статьи: для переформулировки условия равенства $P = NP$ нужен один из результатов, ранее опубликованных автором в статье [3], – однако *мы в настоящее время не считаем этот результат доказанным*. Конкретно, употребляя терминологию настоящей статьи, речь идёт о *необходимом* условии выполнения бинарного отношения $A \Leftrightarrow B$.

Приведём конкретные «претензии» к *своей же* статье [3]. Условие (6.2) – оно, конечно же, содержит опечатку, имелось в виду следующее:

$$A^* \equiv \infty \mathcal{D} \cdot \mathcal{D}^*;$$

заметим, что мы *не можем* переписать это условие с использованием терминологии настоящей статьи, т.е. бинарного отношения \Leftrightarrow . Имеется также опечатка в формулировке леммы 6: исходным условием должно быть $v \in A^*$.

(Стоит отметить, что \mathcal{D} – это некоторый *предполагаемый* язык, *несуществование* которого мы и пытались доказать в статье [3].)

Однако, конечно, обе приведённые выше «претензии» ошибками не являются. Конкретную же ошибку мы указывать не будем (её объяснять долго, сложно и незачем) – но вместо этого отметим *наличие контрпримера* к доказываемому в [3, Th. 1] условию $\mathcal{D} \Leftrightarrow A$ (здесь мы для удобства чтения снова используем терминологию и обозначения настоящей статьи)¹.

¹ Здесь имеется практически полная – и при этом несколько неожиданная – аналогия с тестированием программного обеспечения (ПО). А именно, согласно разным источникам, тесты ПО создаются:

- либо на основе внутренней структуры кода или алгоритма («белый ящик», «white box testing»);
- либо функциональности объекта тестирования («чёрный ящик», «black box testing»).

Разработка тестов «белым ящиком» позволяет, в частности, проверить логику работы кода, а «чёрным ящиком» – проверить работу ПО на некорректных данных и таким образом тестировать его «с точки зрения пользователя».

Итак, необходимый нам контрпример (согласно [4] – «работа чёрного ящика») может быть получен на основе рассмотрения «усложнения» примера из [5], [6]. А именно, мы следующим образом усложняем пример из [5, раздел IV] (к этому же примеру мы возвращались также в [6]). Вместо рассматривавшихся там языков

$$A = \{aaa, aabba, abba, bb\}$$

$$\text{и } B = \{aaaa, abb, abba, bbb\}$$

будем для заданных таким же способом языков A и B рассматривать «существенно более громоздкие» языки

$$A' = (A \cup B)^2 \quad \text{и} \quad B' = (A \cup B)^3. \quad (7)$$

(Отметим ещё, что рассмотрение этих же языков – без связи с конечными автоматами PRI и NSPRI – было начато в [7], [8]. Определения конечных автоматов мы используем в соответствии с [9], [10]. См. также в [11] краткое описание соответствующей компьютерной программы построения автоматов PRI и NSPRI – с помощью которой можно подробно исследовать приведённые в рассматриваемом примере «громоздкие» языки; однако далее будет понятно, что можно обойтись и без такого подробного исследования.)

При выбранном ранее задании «входных данных», т. е. (7), понятно, что:

- во-первых, получающееся множество языков, являющихся состояниями автомата $PRI(A', B')$, очень велико; вряд ли его стóит полностью приводить в какой-либо статье;
- во-вторых, это множество языков включает в качестве подмножества все слова из [5, таб. 1]².

Поэтому, с одной стороны, для пары языков (7) это получающееся множество языков включает, среди прочих, язык $\{a\}$ – и, следовательно, само слово a является одним из слов языка \mathcal{D} , построенного для этой пары языков в доказательстве [3, Th. 1]. (Про построение языка, фактически являющегося языком \mathcal{D} в префиксном случае, – т. е. при рассмотрении только префиксных морфизмов максимальных префиксных кодов³ – см. также в [12].)

Но, с другой стороны (мы возвращаемся к рассматриваемому примеру), эти два языка (7), конечно же, удовлетворяют бинарным отношениям

$$(A \cup B)^2 \not\subseteq (A \cup B)^3 \not\subseteq (A \cup B);$$

но при этом условие

$$(A \cup B) \not\subseteq (A \cup B \cup \{a\})$$

не выполняется: ни для какого $k \geq 1$ слово a^k в язык $(A \cup B)^*$ не входит. Однако, на основе рассмотренных в [7], [8], [11] примеров, мы знаем, что слово a (соответственно, язык $\{a\}$) получается в процессе построения автомата NSPRI(A, B) (соответственно, автомата PRI(A, B)) – и поэтому этот же язык $\{a\}$ должен быть включён в множество состояний автомата

$$PRI(A', B') = PRI((A \cup B)^2, (A \cup B)^3).$$

Мы вместо «белого ящика» (сложного объяснения конкретной «неодделки» в вышеупомянутом ошибочном доказательстве) – применяем «чёрный ящик» (указываем конкретный пример, когда одно из приведённых там же условий не выполняется).

² Т. е. просто «старые» слова.

³ А не произвольных морфизмов и/или расширенных максимальных префиксных кодов.

Итак, полученное здесь противоречие как раз и является необходимым контрпримером к доказывавшемуся в [3, Th. 1] утверждению $\mathcal{D} \not\subseteq A$.

В заключение раздела отметим следующее. Пусть теперь зафиксирован некоторый язык B . Чисто формально мы можем считать, что все слова, входящие в множество $\text{osuff}(B)$, принадлежат множеству состояний недетерминированного автомата NSPRI(A, B) – причём для любого⁴ языка A . Следовательно, все подмножества множества таких слов принадлежат множеству состояний детерминированного автомата PRI(A, B).

Но, с другой стороны, в реальных ситуациях мы можем рассматривать существенно меньшее число состояний для обоих этих автоматов: в частности, в наших примерах у языка

$$B = \{aaaa, abb, abba, bbb\}$$

имеются такие собственные суффиксы его слов:

$$a, aa, aaa, b, ba, bb, bba$$

(в алфавитном порядке). Всего 7 слов, поэтому формально можно считать, что число состояний автомата PRI равно

$$2^7 = 128.$$

(Причём, в отличие от большинства подобных ситуаций, вычитать 1 из степени двойки не надо: мы, среди прочих, должны рассматривать и состояние, соответствующее пустому множеству суффиксов. В наших публикациях во всех рассмотренных ранее примерах это состояние всегда имело номер 0. Более того, подобные состояния мы искусственно вводили и для недетерминированных автоматов NSPRI.)

Вернёмся к автомату PRI. Как мы видели в [5], можно рассматривать эквивалентный автомат⁵, обычно имеющий существенно меньшее число состояний. В частности, в рассмотренном там примере число состояний было равно 16 – вместо потенциальных 128. Сам алгоритм построения, также приведённый в [5]⁶, работает только с состояниями, которые впоследствии не окажутся недостижимыми⁷.

⁴ Во всех примерах, приведённых в этом разделе, для строимых автоматов NSPRI(A, B) оба языка – A и B – совпадают. Подобные примеры совпадения этих языков также уже были рассмотрены в наших предыдущих статьях, процитированных выше.

⁵ Он, конечно, также был назван PRI.

⁶ Варианты этого алгоритма были приведены и в других процитированных выше статьях.

⁷ В качестве комментария к «появлению» языков, являющихся состояниями автомата PRI, отметим такую полную аналогию. «Студенческий» алгоритм детерминизации конечного автомата возможен в двух таких вариантах.

Во-первых (описывается этот вариант проще, но работа алгоритма более сложная) – заранее рассматриваются все возможные подмножества множества состояний исходного недетерминированного автомата, на них естественным образом строится новая функция переходов, и в конце работы алгоритма из полученного автомата удаляются недостижимые состояния. (Далее обычно проводится канонизация полученного детерминированного автомата – но мы сейчас о ней не говорим.)

Во-вторых – мы начинаем с единственного входного состояния, и далее итерационно на всех уже имеющихся состояниях таким же способом определяется функция переходов; при этом недостижимых состояний не возникает, алгоритм описывается сложнее, но в среднем, конечно, работает быстрее.

Такие же два варианта возможны и для рассматриваемого нами алгоритма построения автомата PRI.

XIV. ОБ ОДНОЙ ГИПОТЕЗЕ
ТЕОРИИ ФОРМАЛЬНЫХ ЯЗЫКОВ –
ВАРИАНТЫ ФОРМУЛИРОВКИ

... Итак, мы привели контрпример к одному из вспомогательных доказательств, приведённых в [3]. Однако важно отметить, что это – действительно *контрпример к приведённому доказательству, но вовсе не к доказываемому там утверждению*⁸.

Всё это позволяет сформулировать рассматривавшееся утверждение *в виде гипотезы* – и далее в этом разделе такая формулировка приведена. Мы считаем, что оно (такое утверждение) является очень важной гипотезой⁹ *всей теории формальных языков*. При выполнении этой гипотезы мы сможем утверждать, что специальное подмножество множества языков, *каждый из которых является инверсным морфическим образом заданного языка*, образует не только полурешётку по объединению¹⁰, но и полурешётку по пересечению (и, следовательно, *решётку*)¹¹.

Гипотеза \mathfrak{E} .

Для любых конечных языков $A, B \subseteq \Sigma^*$ выполнено следующее. Эквивалентность $A \Leftrightarrow B$ выполняется тогда и только тогда, когда существует язык $D \subseteq \Sigma^*$, такой что при рассмотрении соответствующего алфавита Δ (где $|\Delta| = |D|$) существуют расширенные максимальные префиксные коды¹²

$$A_\Delta, B_\Delta \subseteq \Delta^*,$$

такие что

$$A = h_D(A_\Delta), \quad B = h_D(B_\Delta).$$

Будем кратко обозначать эту гипотезу¹³ записью (\mathfrak{E}) . \square

⁸ Совершенно не изменит («не ухудшит» и «не улучшит») ситуацию, если мы вместо «чёрного ящика», т.е. контрпримера, ответим на этот же вопрос с помощью «белого ящика» – т.е. укажем конкретную претензию к доказательству. Повторим, что всё это долго, сложно и для целей настоящей статьи ненужно – но, возможно, станет предметом отдельной публикации в будущем.

⁹ На самом деле гипотезы две, они ниже обозначены (\mathfrak{E}) и (\mathfrak{E}') . Вторая является «упрощённым вариантом» первой.

¹⁰ Существование полурешётки по объединению для задачи построения инверсного морфизма очевидно в любой естественной модели.

¹¹ Более того, в статье-продолжении мы предполагаем привести *полиномиальный* алгоритм построения оптимального инверсного морфизма – необходимый, в частности, для *переформулировки* условия равенства $P = NP$. Этот полиномиальный алгоритм возможен в случае выполнения сформулированной в этом разделе гипотезы.

¹² Кратко повторим соответствующие определения, подробнее см. в части I.

Максимальный префиксный код над алфавитом Δ – это некоторый язык над этим алфавитом, являющийся префиксным и имеющий кодовый индекс, равный 1.

Иными словами: некоторый язык $A_\Delta \subseteq \Delta^*$ является *максимальным префиксным кодом*, если для любого ω -слова α над алфавитом Δ существует, причём единственное, слово $u_\Delta \in A_\Delta$, такое что $u_\Delta \in \text{pref}(\alpha)$. При этом вместо всех ω -слов можно рассматривать только все слова языка Δ^k – для какого-либо

$$k \geq \max\{|u_\Delta| \mid u_\Delta \in A_\Delta\}.$$

Некоторый язык над Δ является *расширенным максимальным префиксным кодом* над алфавитом Δ тогда и только тогда, когда существует его непустое подмножество, являющееся максимальным префиксным кодом над этим алфавитом. (В качестве альтернативного определения можно из определения предыдущего убрать слова «причём единственное».)

¹³ В процитированных выше статьях эта гипотеза «доказывалась как теорема». Контрпример был приведён в настоящей статье, в разделе XIII.

Приведём несколько *эквивалентных формулировок* этой же гипотезы (\mathfrak{E}) ; сама эквивалентность следует из изложенного в настоящей статье материала.

Гипотеза \mathfrak{E}' .

Пусть задан некоторый конечный язык $D \subseteq \Sigma^*$. При этом D – минимальный в своём классе эквивалентности¹⁴; это означает¹⁵, что не существует языка $\mathcal{D} \subseteq \Sigma^*$, такого что $D \in \text{tp}^+(\mathcal{D})$.

Любой конечный язык $A \subseteq \Sigma^*$, принадлежащий рассматриваемому классу эквивалентности $A \Leftrightarrow D$, может быть построен за следующие 4 шага:

- 1) выбирается некоторый алфавит Δ , такой что $|\Delta| = |D|$;
- 2) строится некоторый язык $A'_\Delta \in \text{tp}(D)$;
- 3) строится некоторый (любой – но, конечно, зависящий от исходного языка A) язык $A_\Delta \subseteq \Delta^*$, такой что $A_\Delta \supseteq A'_\Delta$; т.е., неформально, к языку A'_Δ добавляются произвольные слова, а формально – $A_\Delta \in \text{tp}^+(\Delta)$;
- 4) строится язык $h_D(A_\Delta)$.

Построенный язык должен совпасть с языком A . \square

Гипотеза \mathfrak{E}'' .

Утверждение, сформулированной в [3, Th. 1], является верным – т.е. сформулированное в [3] достаточное условие эквивалентности $A \Leftrightarrow B$ является необходимым и достаточным. \square

Теперь сформулируем эквивалентный вариант гипотезы, опираясь на возможное её *не* выполнение. Для этого ответим на вопрос, как удачно сформулировать, что при этом невыполнении «может произойти». По-видимому, лучше всего ответить на этот вопрос на основе формулировки (\mathfrak{E}') . Раз *в потенциальном контрпримере* не все слова языка A принадлежат *минимальному* множеству (языку) D^* , то существует некоторый минимальный в своём классе эквивалентности язык D (будем считать, что именно он задаёт рассматриваемый нами класс эквивалентности) и *не* принадлежащее множеству D^* слово (пусть это $u \in \Sigma^*$), такие что

$$(D \cup \{u\}) \Leftrightarrow D$$

(или, что эквивалентно, $(D \cup \{u\}) \not\leq D$). Поэтому гипотезу формулируем просто как *несуществование* такой пары.

Гипотеза \mathfrak{E}''' .

Не существует пары (D, u) , где

$$D \subseteq \Sigma^* \quad \text{и} \quad u \in \Sigma^*, \quad u \notin D^*,$$

такой что:

- язык D – минимальный в своём классе эквивалентности¹⁶;
- $(D \cup \{u\}) \Leftrightarrow D$. \square

¹⁴ При формулировке основного варианта данной гипотезы, т.е. (\mathfrak{E}) , такого дополнительного требования не нужно.

¹⁵ Возможно также объяснение минимальности с помощью одного из описанных в разделе XI настоящей статьи (часть II) частичных порядков на множестве языков. Более того, «эквивалентность для наших задач различных вариантов минимальности» – возможная тема отдельной публикации.

¹⁶ Об этой минимальности – см. комментарий и сноску в варианте формулировки (\mathfrak{E}') .

В оставшейся части статьи мы будем в первую очередь рассматривать последний вариант гипотезы – т. е. (\mathbb{X}''') . Понятно, что мы *не знаем* ни одной подобной пары (u, D) – и также не знаем доказательства несуществования такой пары (а иначе всё это нельзя было бы называть гипотезой); однако сейчас мы *предположим, что такая пара существует* – и будем рассматривать соответствующие язык D (ещё раз повторим, что он – минимальный в своём классе эквивалентности) и не принадлежащее языку D^* слово u .

В рамках сделанных предположений несложно убедиться в следующем факте – формулируемом далее в этом абзаце. Пусть v – слово языка D (причём любое слово). Тогда «уберём» из D это слово v и «вместо него добавим» ранее отмеченное слово u – обозначим полученный язык D_{u-v} . Тогда неверно, что

$$D \not\subseteq D_{u-v}. \quad (8)$$

(По-видимому, лучше говорить «менее обязывающе», т. е. что «это, вообще говоря, неверно» – потому что соответствующие примеры у нас фактически уже были, а доказательства в общем случае мы пока не приводим. Например, для $v = a$ – или, вообще говоря, для любого 1-буквенного слова – мы не можем ни для какого $k \geq 2$ иметь слово a^k

- ни в качестве другого элемента языка D ,
- ни в качестве слова u –

поскольку оба этих варианта противоречили бы сделанным предположениям. Однако и для произвольного слова v , не обязательно 1-буквенного, соответствующее доказательство провести возможно¹⁷.)

Таким образом, вообще говоря, условие (8) неверно. Однако из этого факта вовсе не следует

- ни возможность,
- ни невозможность

существования нескольких «нарушений» рассматриваемой гипотезы (\mathbb{X}) ¹⁸ – причём для одного и того же языка D (это важно для дальнейшего). Более того, мы не можем утверждать, что невозможна ситуация, когда «замена» нескольких слов языка D (причём не менее чем 1 слова) на несколько других слов¹⁹ даёт язык, принадлежащий классу эквивалентности по отношению $\not\subseteq$, – классу, образованному рассматриваемым языком D . Эквивалентность, аналогичную (8), при этом логично записать так:

$$D \not\subseteq D_{F-G}$$

(здесь G – удаляемое множество слов, а F – добавляемое множество).

Сформулируем всё сказанное в виде следующей гипотезы (\mathbb{X}'') .

Гипотеза \mathbb{X}'' .

Гипотеза (\mathbb{X}) не выполняется. Однако при этом *не существует* конечного языка $D \subseteq \Sigma^*$, для которого выполнено следующее.

¹⁷ Это доказательство вряд ли представляет интерес для настоящей статьи – но может быть частью предмета одной из будущих публикаций.

¹⁸ Повторим, это не связано с условием (8).

¹⁹ Каждое из которых *нарушает* гипотезу (\mathbb{X}''') в качестве слова u – и, следовательно, основную формулировку, гипотезу (\mathbb{X}) .

Во-первых, для этого D гипотеза (\mathbb{X}) *не выполняется* для каждого из следующих слов:

$$u_1, u_2, \dots, u_n$$

(согласно приведённым выше пояснениям, выполнено не только неравенство $n \geq 1$, но и $n \geq 2$; кроме того, ни одно из этих слов не входит в язык D^*). Обозначим

$$\{u_1, u_2, \dots, u_n\} = F.$$

Во-вторых, для этих языков D и F существует некоторое непустое множество $G \subseteq D$, такое что

$$D \not\subseteq D_{F-G}.$$

Будем кратко обозначать эту гипотезу записью (\mathbb{X}'') . \square

Включения двух сформулированных гипотез можно изобразить на следующем рис. 1:



Рис. 1. Включение гипотез.

Здесь каждой «точке» рисунка соответствует некоторый язык²⁰ – и, конечно же,

- как «внутреннее» множество языков, соответствующее потенциальному выполнению гипотезы (\mathbb{X}) ,
- так и «внешнее» множество языков, соответствующее потенциальному невыполнению гипотезы (\mathbb{X}) , но при этом выполнению гипотезы (\mathbb{X}'') –

не могут быть пустыми (что очевидно в связи с существованием многих тривиальных примеров²¹) – но при этом на самом деле могут совпадать с «внешними для них» множествами языков. В таком возможном совпадении множеств языков и заключаются сами эти гипотезы.

XV. ОБ ОДНОЙ ГИПОТЕЗЕ

ТЕОРИИ ФОРМАЛЬНЫХ ЯЗЫКОВ – ЕЩЁ РАЗ О БЕСКОНЕЧНЫХ ИТЕРАЦИОННЫХ ДЕРЕВЬЯХ

Сформулируем приведённую выше гипотезу в терминах бесконечных итерационных деревьев (БИДов), рассматривавшихся нами в [5], [6], [7], [8]. Будем применять вариант гипотезы (\mathbb{X}''') . Конечно, здесь мы строим некоторое конкретное БИД для *одного* языка (в некоторых моделях, рассмотренных в этих публикациях, «на входе» была пара конечных языков) – а именно того, который во всех формулировках гипотезы обозначался как D ; мы рассматриваем его итерации.

²⁰ Точнее – пара (язык D , слово u) для первой гипотезы и тройка (язык D , язык F , язык G) для второй гипотезы. На рисунке мы в обоих случаях имеем в виду только варианты языка D .

²¹ Достаточно хотя бы рассмотреть язык $D = \{a, b\}$ над алфавитом $\Sigma = \{a, b\}$. (Упрощая, можно говорить, что «здесь язык совпадает с алфавитом».) Конечно же, никакого «плохого» слова u в этом примере не существует – просто потому, что здесь не существует слова, не принадлежащего Σ^* .

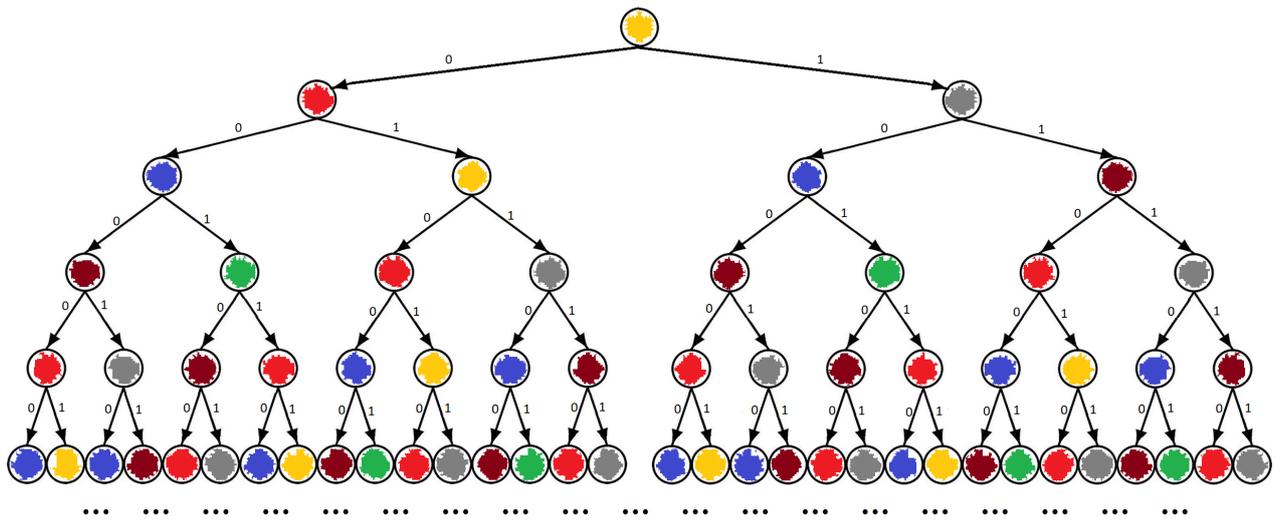


Рис. 2. Повтор [8, рис. 14] – пример абстрактного бесконечного итерационного дерева морфизма для алфавита $\Delta = \{0, 1\}$ (показаны уровни от 0-го до 5-го).

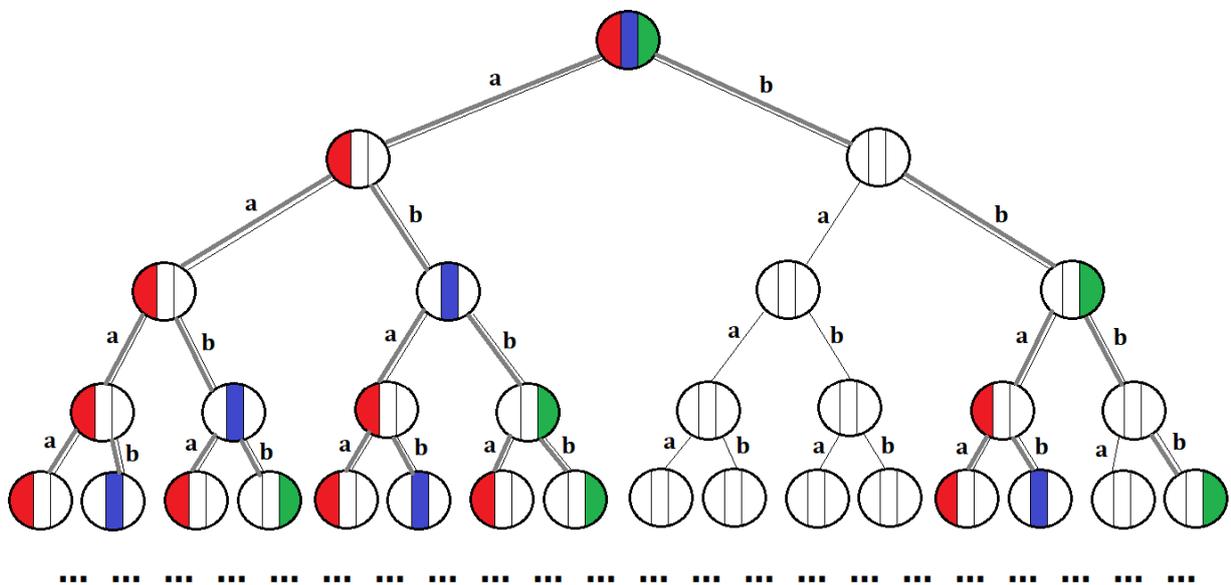


Рис. 3. Пример конкретного бесконечного итерационного дерева для алфавита $\Sigma = \{a, b\}$ и языка $\{a, ab, bb\}$ (показаны уровни дерева от 0-го до 4-го).

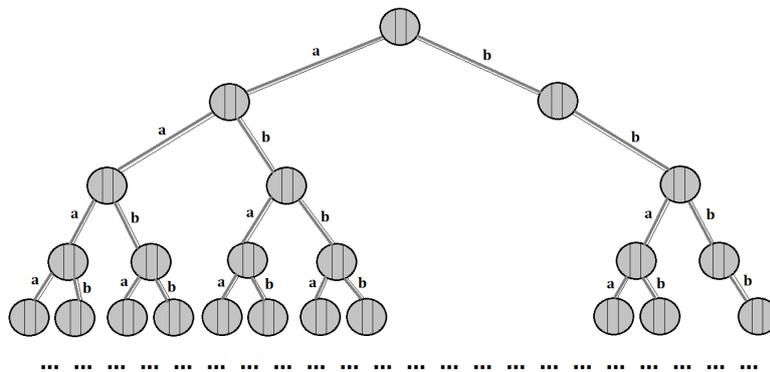


Рис. 4. Пример бесконечного итерационного дерева, существование которого для невыполнения гипотезы (\bar{A}) было бы нужно в примере с предыдущего рисунка – причём с корнем в какой-либо неотмеченной там вершине.

Сразу отметим следующее обстоятельство. Во всех примерах, рассмотренных в процитированных статьях, деревья строились над тем алфавитом, который мы практически всегда обозначаем Δ ; «наиболее интересные» из рассмотренных примеров таких деревьев были приведены на следующих рисунках: [8, рис. 14] и впоследствии [5, рис. 1]. «Самый интересный» и при этом «наиболее абстрактный» [8, рис. 14] мы повторяем здесь – это рис. 2 настоящей статьи.

Что там нарисовано?²² Каждое слово над рассматриваемым алфавитом Δ можно прочесть на (единственном для него) пути, начинающемся в корне дерева. На рисунке одним и тем же цветом отмечены вершины (слова), относящиеся к одному и тому же классу эквивалентности; пример сделан таким образом, что (бесконечные) поддерева, относящиеся к вершинам одного класса, также эквивалентны²³ – в противном случае нельзя было бы говорить, что рассматривается БИД.

Повторим, что рассмотренный пример – «абстрактный», т. е. какого-либо конкретного морфизма, для которого получается именно приведённое на рис. 2 дерево, мы не даём²⁴. Хотя, конечно, несмотря на рассмотрение в предыдущих статьях слов над алфавитом Δ , нас существенно больше интересуют морфизмы самих этих слов – слова над алфавитом Σ .

На аналогичном бесконечном дереве – но, однако, построенном для слов алфавита Σ – будем отмечать слова, принадлежащие языку D^* ; для примера рассмотрим язык $D = \{a, ab, bb\}$. На рис. 3 помечаем слова языка D^* следующим образом:

- красным цветом – слова, (единственное) разложение которых на слова языка D кончается на слово a ;
- синим цветом – аналогично для слова ab ;
- зелёным цветом – аналогично для слова bb .

Понятно, что в принципе некоторое слово может быть помечено более чем одним цветом (тривиальный пример получается при наличии в языке слов a и aa одновременно). Кроме того, по смыслу таких пометок, желательно также пометить всеми возможными цветами корень дерева, вершину ϵ .

Про полученное дерево – отметим, что его тоже можно назвать БИДом – сделаем такие два «не очень важных» замечания²⁵.

- Во-первых, для заданного таким образом языка D выполнен критерий однозначности декодирования – несмотря на то, что D не обладает свойством префикса²⁶. В связи с этим ни одна вершина, соответствующая слову языка Σ^* , не «покрашена» в 2 (и

²² Мы совершенно по-иному объясняем материал и рисунки той статьи.

²³ Сама эквивалентность бесконечных деревьев определяется естественным образом.

²⁴ Это можно рассматривать как отдельную задачу – построение конкретного морфизма по заданному БИД. Решение этой задачи может быть полезным для получения инъективного отображения множества БИДов во множество морфизмов. (При этом инъективное отображение «в обратную сторону», т. е. отображение множества морфизмов во множество БИДов, строится алгоритмически – и очень похожею задачу мы фактически рассматривали при построении автомата PRI.)

²⁵ Точнее, эти замечания мало связаны с предметом настоящей статьи.

²⁶ Самое простое объяснение этого – D обладает свойством суффикса.

тем более в 3) цвета одновременно – кроме, конечно, корня дерева, вершины ϵ .

- Во-вторых, любое слово языка Σ^* , имеющее префикс a , «покрашено». Это для рассматриваемого языка D можно доказать строго.

Существенно важнее следующее. Мы помечаем пути этого дерева, соответствующие всем «покрашенным» словам: на рис. 3 помечаем рёбра каждого из этих путей серым цветом; для удобства рассмотрим упрощение рисунка 3, т. е. рис. 4, – на нём мы оставили только эти рёбра. Гипотеза (\mathfrak{A}) (более точно, её версия (\mathfrak{A}''')) «с точки зрения БИДов» заключается в том, что *в таком бесконечном итерационном дереве для алфавита Σ не существует ни одной такой «непокрашенной» вершины, что БИД, соответствующее этой вершине как корню, включает в себя рассматриваемое «серое» БИД в качестве подмножества*²⁷.

Понятно, что в рассматриваемом нами примере такие вершины могли бы быть только в «правом» поддереве – т. е. только среди слов, начинающихся с b . Вообще, для рассматриваемого примера можно строго доказать что такого БИДа не существует. Это удаётся доказать и для всех примеров, рассмотренных нами – как «вручную»²⁸, так и с помощью компьютерных программ (описанных в [7], [8], [11] и др.); а гипотеза (\mathfrak{A}) заключается именно в том, что *этот факт выполняется всегда*.

В заключение раздела отметим следующее. Кроме формулировки в терминах бесконечных итерационных деревьев приведённую гипотезу (\mathfrak{A}) можно переформулировать и в терминах недетерминированных конечных автоматов (прежде всего – с помощью определённого в разделе II части I автомата \mathcal{K} для заданного конечного языка, в нашем случае – $\mathcal{K}(D)$). Немного подробнее см. уже процитированные статьи [5], [6], а также [13], [14], [15] – однако более подробно эту тему мы предполагаем рассмотреть в следующих публикациях.

XVI. РЕШЁТКА В ЗАДАЧЕ ПОСТРОЕНИЯ ОПТИМАЛЬНОГО ИНВЕРСНОГО МОРФИЗМА

В этом разделе мы показываем, что на множестве Ω подмножеств потенциальных корней (точнее – на специальном подмножестве множества Ω) при выполнении сформулированной выше гипотезы (\mathfrak{A}) выполняется определение решётки по пересечению и объединению.

Конечно, про «самую обычную» полурешётку по пересечению (в задаче построения оптимального инверсного морфизма) мы говорить не можем: например, для языка $D = \{a\}$ можно рассмотреть такой тривиальный пример:

$$A = \{a^2\}, \quad B = \{a^3\};$$

для него

$$D \not\subseteq A \not\subseteq B, \quad \text{и при этом } A \cap B = \emptyset.$$

В связи с этим сделаем такое *первое упрощение*. Будем рассматривать только те потенциальные корни, которые *не представимы* в виде конкатенации других потенциальных корней: для исходного множества потенциальных

²⁷ Здесь приведено неформальное изложение этой гипотезы, использующее слова «непокрашенный», «серый» и т. п. – но, конечно, всё это очень легко записать «в виде формул».

²⁸ Приведённый в одной из предыдущих сносок язык $D = \{a, b\}$ и является одним из таких примеров, причём тривиальным.

корней $\Omega' \subseteq \Sigma^*$, относящемуся к заданному конечному языку $A \subseteq \Sigma^*$, т. е.

$$\Omega' = \{ u \in \Sigma^* \mid (\exists k \in \mathbb{N}) (u^k \in A) \},$$

будем рассматривать такое его подмножество $\Omega \subseteq \Sigma^*$:

$$\begin{aligned} \Omega &= \Omega' \setminus \Omega'', \text{ где} \\ \Omega'' &= \{ u \in \Omega' \mid (\exists u_1, u_2, \dots, u_k \in \Omega', k \geq 2) \\ &\quad (u_1 u_2 \dots u_k = u) \}. \end{aligned}$$

Сразу сделаем такое замечание. Казалось бы, такое же упрощение можно сделать и в первой задаче?²⁹ Однако там это гораздо менее интересно: в первой задаче мы все потенциальные корни получаем путём «деления слов на конкретную, заранее заданную константу», – и именно поэтому там представляет интерес работа *только со всеми* потенциальными корнями.

Теперь отметим, что при предположении о выполнении гипотезы (\mathbb{A}) можно сделать вывод, что полурешётка состоит из единственного множества – выше обозначенного нами Ω (напомним, что мы рассматриваем полурешётку по пересечению). Объяснить это можно следующим образом: согласно (\mathbb{A}), любое слово языка, входящего в класс эквивалентности, образованный некоторым минимальным языком D (а последним, как несложно убедиться, при выполнении гипотезы (\mathbb{A}) по каждой рассмотренной выше метрике и является язык Ω), представляется как некоторая конкатенация слов D (или Ω). Таким образом, при сделанном выше *первом упрощении* полурешётка по пересечению, конечно, имеется (а следовательно, имеется и решётка), – хотя состоит эта полурешётка *из единственного множества* потенциальных корней – из всего Ω .

Поэтому вызывает интерес рассмотрение полурешётки (по тому же пересечению) не для отношения \trianglelefteq , а для отношения \triangleright . Итак, мы начинаем использовать такое *второе упрощение*. В нём рассмотрение отношения \triangleright означает, что в качестве элементов полурешётки выбираются те подмножества множества Ω , которые находятся в этом отношении \triangleright с множеством Ω : т. е. такие A , что

$$A \triangleright \Omega$$

(понятно, что при этом мы рассматриваем только такие A , что $A \subseteq \Omega'$).

«Зато» при этом мы далее будем считать, что *выполнено менее сильное утверждение* – (\mathbb{A})³⁰. По-видимому, это можно назвать *третьим упрощением* (а не усложнением) – или ещё одним условием существования полурешётки.

²⁹ Т. е. в задаче извлечения корня из языка.

³⁰ Неформально: без этого утверждения – о какой полурешётке можно говорить? Объясним это, также *неформально*.

Если не выполнена гипотеза (\mathbb{A}), то для рассматриваемого минимального языка D обязательно существует эквивалентный ему по отношению \trianglelefteq язык, который *не содержит* некоторые слова языка D (поскольку, согласно сделанному предположению и в терминых введённых выше обозначений, $G \neq \emptyset$); пусть это язык D' .

Понятно, что

$$D \cap D' \subseteq D$$

(т. е. не равен D), поэтому, вследствие предположенной нами минимальности языка D мы получаем, что

$$D \cap D' \triangleleft D -$$

и условие для полурешётки по пересечению не выполняется.

При всех сделанных предположениях существование полурешётки по пересечению очевидно. Действительно, пересечение любых таких множеств (элементов формируемой полурешётки) по построению содержит язык Ω и поэтому действительно является полурешёткой³¹.

При этом важно отметить, что «такое бывает не всегда»; приведём только один пример из наших предыдущих построений. Подобное условие для полурешётки по пересечению не выполняется (вообще говоря) для нескольких различных задач, возникающих при исследовании универсального автомата (the universal automaton, [16], [17], [18], [19] и др.; также см. статьи про полный автомат, the complete automaton, [20] и др.). Конкретно, при построении различных вариантов покрывающего автомата может не существовать единственного «минимального» множества гридов (блоков), и наиболее простой поясняющий пример, по-видимому, такой³².

#	#	#	
#	#		#
#		#	#
	#	#	#

1	1	1	0
1	1	0	1
1	0	1	1
0	1	1	1

Рис. 5. Пример отношения # для регулярного языка.

На приведённом рис. 5 показано отношение # (см. [9] и др.) для некоторого регулярного языка (конечного автомата) – слева и справа в разных форматах, оба формата используются в наших публикациях. (Отметим, что согласно [20] для любого «приемлемого» варианта отношения # существует соответствующий ему регулярный язык³³.) Возможны 2 варианта, соответствующих минимальным автоматам – «по строкам» (получается канонический автомат, который мы обычно обозначаем \tilde{L}) и «по столбцам» (автомат $(\tilde{L}^R)^R$). Но два множества гридов – в первом случае (описывающие строки) и во втором случае (описывающие столбцы) – в пересечении дают пустое множество, которое, конечно же, не описывает эквивалентный автомат; поэтому приведённая конструкция (множество подмножеств множества гридов) не интерпретируется как полурешётка.

И, конечно, вместе с операцией объединения мы получаем решётку: объединение таких множеств потенциальных корней A_1 и A_2 , что

$$A_1 \triangleright \Omega \text{ и } A_2 \triangleright \Omega,$$

даёт такое множество A , что так же

$$A \triangleright \Omega.$$

³¹ На самом деле условия идемпотентности, ассоциативности и коммутативности для операции пересечения выполняются по определению – и поэтому в нашей ситуации строимая алгебра *не являлась бы* полурешёткой, если бы пересечение каких-либо двух объектов не являлось бы элементом предполагаемой полурешётки; мы здесь как раз и показываем, что *в нашем случае такое невозможно*.

И, конечно, включение «лишних» потенциальных корней в рассматриваемом нами случае (т. е. при всех сделанных выше упрощениях) не является проблемой: мы ведь рассматриваем те элементы A формируемой полурешётки, для которых выполнено условие $A \triangleright \Omega$ – не обязательно $A \trianglelefteq \Omega$.

³² Интересно отметить, что тот же самый пример мы обычно приводим и для совершенно иной задачи, связанной с недетерминированными конечными автоматами, – а именно, с описанием эвристических алгоритмов их минимизации.

³³ В процитированных статьях описаны необходимые для этого алгоритмы – т. е. доказательство существования является конструктивным.

То есть множество-объединение, во-первых, состоит из элементов – потенциальных корней, и, во-вторых, также удовлетворяет требуемым ограничениям.

В заключение раздела снова сделаем замечание, совершенно аналогичное приведённому после формулировки первого упрощения. Казалось бы, все приведённые нами упрощения можно сделать и в первой задаче? И ответ примерно такой же: там, в первой задаче, представляет интерес работа *только со всеми* потенциальными корнями. Однако, конечно, и в задаче извлечения корня из языка при рассмотрении отношения \triangleright (вместо \trianglelefteq) мы тоже получаем полурешётку по объединению.

XVII. ЗАКЛЮЧЕНИЕ

В этом разделе мы, в отличие от большинства наших статей, приводим очень мало проблем для дальнейшего исследования: такие проблемы уже были сформулированы ранее по тексту статьи. Более того, отметим сразу, что весь раздел VII (часть II) можно считать совокупностью таких формулировок. Поэтому среди предполагаемых направлений будущей работы перечислим только самые важные.

Автор собирается опубликовать *конкретную* неточность в одном из вспомогательных доказательств, ранее приведённых в [3]. Также важно рассмотрение «эквивалентности» для наших задач различных вариантов минимальности языка (обычно обозначаемого нами D) – это должно упростить формулировки результатов и доказательства, использующие такой минимальный язык.

В самом конце статьи отметим, что в одной из следующих публикаций мы собираемся подробно рассмотреть автоматы, которые в тексте настоящей статьи были названы автоматами $\mathcal{K}(A)$ для заданного конечного языка $A \subseteq \Sigma^*$. Одно из названий подобных автоматов в англоязычной литературе – semi-flower automata, [21]; по-видимому, удачное русское название, уже применённое в одной из сносок части I, – «лепестковые автоматы».

Эти автоматы несколько раз были использованы в нашей статье – поэтому их подробное рассмотрение должно быть интересно для дальнейшего исследования рассмотренных выше вопросов. Прежде всего мы собираемся привести доказательство существования лепесткового автомата, соответствующего любой возможной таблице бинарного отношения $\#$ (т.е., иными словами, – для таблицы, соответствующей любому регулярному языку). Далее – доказательство существования такого автомата для любой последовательности состояний в такой таблице (поскольку для состояний лепесткового автомата можно естественным образом ввести частичный порядок – что даст варианты полного линейного порядка для автомата, соответствующего таблице бинарного отношения $\#$). Всё это даст возможность свести к рассмотрению регулярных языков и недетерминированных конечных автоматов (причём не только автоматов $PR1$ и $NSPR1$, но и автоматов «более информативных») любой частный случай проблемы проверки гипотезы $A \trianglelefteq B$.

Список литературы

- [1] Мельников Б. *Полурешётки подмножеств потенциальных корней в задачах теории формальных языков. Часть I. Извлечение корня из языка* // International Journal of Open Information Technologies. – 2022. – Vol. 10. No. 4. – P. 1–9.
- [2] Мельников Б. *Полурешётки подмножеств потенциальных корней в задачах теории формальных языков. Часть II. Построение инверсного морфизма* // International Journal of Open Information Technologies. – 2022. – Vol. 10. No. 5. – P. 1–8.
- [3] Melnikov B. *The equality condition for infinite catenations of two sets of finite words* // International Journal of Foundation of Computer Science. – 1993. – Vol. 4. No. 3. – P. 267–274.
- [4] Кулаков К., Димитров В. *Основы тестирования программного обеспечения*. – Петрозаводск, Изд-во Петрозаводского государственного университета. – 2018. – 57 с.
- [5] Мельников Б. *Варианты конечных автоматов, соответствующих бесконечным итерационным деревьям морфизмов. Часть I* // International Journal of Open Information Technologies. – 2021. – Vol. 9. No. 7. – P. 5–13.
- [6] Мельников Б. *Варианты конечных автоматов, соответствующих бесконечным итерационным деревьям морфизмов. Часть II* // International Journal of Open Information Technologies. – 2021. – Vol. 9. No. 10. – P. 1–8.
- [7] Мельников Б., Мельникова А. *Бесконечные деревья в алгоритме проверки условия эквивалентности итераций конечных языков. Часть I* // International Journal of Open Information Technologies. – 2021. – Vol. 9. No. 4. – P. 1–11.
- [8] Мельников Б., Мельникова А. *Бесконечные деревья в алгоритме проверки условия эквивалентности итераций конечных языков. Часть II* // International Journal of Open Information Technologies. – 2021. – Vol. 9. No. 5. – P. 1–11.
- [9] Мельников Б. *Регулярные языки и недетерминированные конечные автоматы (монография)*. – М., Изд-во Российского государственного социального университета. – 2018. – 179 с. – ISBN 978-5-7139-1355-7.
- [10] Melnikov B., Vakhitova A. *Some more on the finite automata* // The Korean Journal of Computational and Applied Mathematics (Journal of Applied Mathematics and Computing). – 1998. – Vol. 5. No. 3. – P. 495–505.
- [11] Абрамян М., Мельников Б. *Алгоритмы преобразования конечных автоматов, соответствующих бесконечным итерационным деревьям* // Современные информационные технологии и ИТ-образование. – 2021. – Том 17. № 1. – С. 13–23.
- [12] Мельников Б. *Подклассы класса контекстно-свободных языков (монография)*. – М., Изд-во Московского университета. – 1995. – 174 с. – ISBN 5-211-03448-1.
- [13] Алексеева А., Мельников Б. *Итерации конечных и бесконечных языков и недетерминированные конечные автоматы* // Вектор науки Тольяттинского государственного университета. – 2011. – № 3 (17). – С. 30–33.
- [14] Коробельщикова С., Мельников Б. *Максимальные префиксные коды и подклассы класса контекстно-свободных языков* // Вестник Северного (Арктического) федерального университета. Серия: Естественные науки. – 2015. – № 1. – С. 121–129.
- [15] Коробельщикова С., Мельников Б. *Итерации языков и максимальные префиксные коды* // Вестник Воронежского государственного университета. Серия: Физика. Математика. – 2015. – № 2. – С. 106–120.
- [16] Lombardy S., Sakarovitch J. *The universal automaton* // Logic and Automata. Amsterdam University Press, 2008, P. 457–504.
- [17] Долгов В., Мельников Б. *Построение универсального конечного автомата. I. От теории к практическим алгоритмам* // Вестник Воронежского государственного университета. Серия: Физика. Математика. – 2013. – № 2. – С. 173–181.
- [18] Долгов В., Мельников Б. *Построение универсального конечного автомата. II. Примеры работы алгоритмов* // Вестник Воронежского государственного университета. Серия: Физика. Математика. – 2014. – № 4. – С. 78–85.
- [19] Melnikov B., Dolgov V. *Some more algorithms for Conway's universal automaton* // Acta universitatis sapientiae informatica. – 2014. – Vol. 6. No. 1. – P. 5–20.
- [20] Melnikov B. *The complete finite automaton* // International Journal of Open Information Technologies. – 2017. – Vol. 5. No. 10. – P. 9–17.
- [21] Singh Sh., Krishna K. *On syntactic complexity of circular semi-flower automata* // Implementation and Application of Automata, CIAA 2018, Lecture Notes in Computer Science, 10977, Springer International Publishing Ag. – 2018. – P. 312–323.

Борис Феликсович МЕЛЬНИКОВ,
 профессор Университета МГУ – ППИ в Шэньчжэне
 (<http://szmsubit.ru/>),
 email: bf-melnikov@yandex.ru,
 mathnet.ru: personid=27967,
 elibrary.ru: authorid=15715,
 scopus.com: authorId=55954040300,
 ORCID: orcidID=0000-0002-6765-6800.

Semi-lattices of the subsets of potential roots in the problems of the formal languages theory. Part III. The condition for the existence of a lattice

Boris Melnikov

Abstract—In the paper, we consider all possible subsets of the set of potential roots forming in some situations semi-lattices, by intersection and/or by union. Such structures arise in two similar problems in the theory of formal languages. Specifically, for some given finite language, we consider the problem of extracting the root of a given degree and the problem of constructing an optimal inverse morphism, where optimality can be defined, for example, as the length of the maximum word of a language that is an inverse morphic image.

In both of the above cases, it is necessary to construct the set of so-called potential roots, i.e., such words of the alphabet in question, for each of which some degree of it is included in the source language. It is important to note that the construction of the set of all potential roots can be performed using a simple polynomial algorithm. Exponential algorithms for both problems are obvious: we just need to sort through all subsets of the set of these potential roots, and choose the appropriate one among these subsets. Therefore, the problem is to describe possible polynomial algorithms for these problems.

For both of these problems, the possible existence of two semi-lattices available on a pre-constructed set of subsets of potential roots is of interest.

Among other things, in the paper we present the formulation of one important hypothesis of the theory of formal languages, in which we can assert that a special subset of a set of languages, each of which is an inverse morphic image of a given language, forms not only a half-lattice by union, but also a half-lattice by intersection (and, therefore, a lattice).

The proposed third part of the article, similar to the second part, is more devoted to the second problem, i.e., the problem of constructing an optimal inverse morphism. We formulate a hypothesis of the theory of formal languages, in which special subsets of the set of potential roots form an intersection semilattice.

Keywords—formal languages, iterations of languages, morphisms, inverse morphisms, binary relations, semi-lattices, algorithms.

References

- [1] Melnikov B. *Semi-lattices of the subsets of potential roots in the problems of the formal languages theory. Part I. Extracting the root from the language* // International Journal of Open Information Technologies. – 2022. – Vol. 10. No. 4. – P. 1–9 (in Russian).
- [2] Melnikov B. *Semi-lattices of the subsets of potential roots in the problems of the formal languages theory. Part II. Constructing an inverse morphism* // International Journal of Open Information Technologies. – 2022. – Vol. 10. No. 5. – P. 1–8 (in Russian).
- [3] Melnikov B. *The equality condition for infinite catenations of two sets of finite words* // International Journal of Foundation of Computer Science. – 1993. – Vol. 4. No. 3. – P. 267–274.
- [4] Kulakov K., Dimitrov V. *Basics of Software Testing*. – Petrozavodsk, Publishing House of Petrozavodsk State University. – 2018. – 57 p. (in Russian).
- [5] Melnikov B. *Variants of finite automata, corresponding to infinite iterative morphism trees. Part I* // International Journal of Open Information Technologies. – 2021. – Vol. 9. No. 7. – P. 5–13 (in Russian).
- [6] Melnikov B. *Variants of finite automata, corresponding to infinite iterative morphism trees. Part II* // International Journal of Open Information Technologies. – 2021. – Vol. 9. No. 10. – P. 1–8 (in Russian).
- [7] Melnikov B., Melnikova A. *Infinite trees in the algorithm for checking the equivalence condition of iterations of finite languages. Part I* // International Journal of Open Information Technologies. – 2021. – Vol. 9. No. 4. – P. 1–11 (in Russian).
- [8] Melnikov B., Melnikova A. *Infinite trees in the algorithm for checking the equivalence condition of iterations of finite languages. Part II* // International Journal of Open Information Technologies. – 2021. – Vol. 9. No. 5. – P. 1–11 (in Russian).
- [9] Melnikov B. *Regular languages and nondeterministic finite automata (monograph)*. – Moscow, Russian Social State University Ed. – 2018. – 179 p. – ISBN 978-5-7139-1355-7. (in Russian).
- [10] Melnikov B., Vakhitova A. *Some more on the finite automata* // The Korean Journal of Computational and Applied Mathematics (Journal of Applied Mathematics and Computing). – 1998. – Vol. 5. No. 3. – P. 495–505.
- [11] Abramyan M., Melnikov B. *Algorithms of transformation of finite automata, corresponding to infinite iterative trees* // Modern information technologies and IT education. – 2021. – Vol. 17. No. 1. – P. 13–23. (in Russian).
- [12] Melnikov B. *Subclasses of the context-free languages class (monograph)*. – Moscow, Moscow State University Ed. – 1995. – 174 p. – ISBN 5-211-03448-1. (in Russian).
- [13] Alekseeva A., Melnikov B. *Iterations of finite and infinite languages and nondeterministic finite automata* // Vector of Science of Togliatti State University. – 2011. – No. 3 (17). – P. 30–33 (in Russian).
- [14] Korabelshchikova S., Melnikov B. *Maximum prefix codes and subclasses of the context-free languages class* // Bulletin of the Northern (Arctic) Federal University. Series: Physics. Mathematics. – Серия: Natural sciences. – 2015. – No. 1. – P. 121–129 (in Russian).
- [15] Korabelshchikova S., Melnikov B. *Iterations of languages and maximum prefix codes* // Bulletin of the Voronezh State University. Series: Physics. Math. – 2015. – No. 2. – P. 106–120 (in Russian).
- [16] Lombardy S., Sakarovitch J. *The universal automaton* // Logic and Automata. Amsterdam University Press, 2008, P. 457–504.
- [17] Dolgov V., Melnikov B. *Constructing universal finite automaton. I. From theory to practical algorithms* // Bulletin of the Voronezh State University. Series: Physics. Mathematics. – 2013. – No. 2. – P. 173–181 (in Russian).
- [18] Dolgov V., Melnikov B. *Constructing universal finite automaton. II. Examples of algorithms* // Bulletin of the Voronezh State University. Series: Physics. Mathematics. – 2014. – No. 4. – P. 78–85 (in Russian).
- [19] Melnikov B., Dolgov V. *Some more algorithms for Conway's universal automaton* // Acta universitatis sapientiae informatica. – 2014. – Vol. 6. No. 1. – P. 5–20.
- [20] Melnikov B. *The complete finite automaton* // International Journal of Open Information Technologies. – 2017. – Vol. 5. No. 10. – P. 9–17.
- [21] Singh Sh., Krishna K. *On syntactic complexity of circular semi-flower automata* // Implementation and Application of Automata, CIAA 2018, Lecture Notes in Computer Science, 10977, Springer International Publishing Ag. – 2018. – P. 312–323.

Boris MELNIKOV,
Professor of Shenzhen MSU–BIT University, China
(<http://szmsubit.ru/>),
email: bf-melnikov@yandex.ru,
[mathnet.ru: personid=27967](http://mathnet.ru/personid=27967),
[elibrary.ru: authorid=15715](http://elibrary.ru/authorid=15715),
[scopus.com: authorId=55954040300](http://scopus.com/authorId=55954040300),
ORCID: [orcidID=0000-0002-6765-6800](http://orcid.org/0000-0002-6765-6800).