Известия вузов. Математика *http://www.ksu.ru/journals/izv_vuz/*
2010, №6, с. 23–31 *Гос. номер статьи по НТЦ "Информрегистр" 0421000123\0054*

Г.А. ПОВАРОВ

КОНЕЧНЫЕ ТРАНСДЬЮСЕРЫ И НЕДЕТЕРМИНИРОВАННАЯ СЛОЖНОСТЬ РЕГУЛЯРНОГО ЯЗЫКА

Аннотация. Изучается недетерминированная сложность применения конечного трансдьюсера к регулярному языку. Получена точная верхняя оценка этой величины.

Ключевые слова: конечный трансдьюсер, недетерминированный конечный автомат, регулярный язык, дескриптивная сложность, недетерминированная сложность.

УДК: 519.713

Abstract. We obtain a sharp upper bound for the nondeterministic state complexity of the application of a finite transducer to a regular language.

Keywords: finite transducer, nondeterministic finite automaton, regular language, descriptive complexity, nondeterministic state complexity.

Введение

Хорошо известна та важная роль, которую играет в теоретических и практических разделах современной информатики понятие *регулярного языка*. В различных ситуациях оказывается удобным использовать разные способы задания регулярных языков — детерминированные и недетерминированные конечные автоматы, право- и леволинейные грамматики, регулярные выражения и др. Для каждого из этих способов уместен вопрос о максимально экономичном задании регулярного языка. Именно такие вопросы ставятся и решаются в рамках теории *дескриптивной сложности* формальных языков, в которой с каждым способом задания языка связывается некоторая числовая мера и исследуется минимально возможное значение этой меры для данного языка. Применительно к регулярным языкам одной из классических мер сложности является *недетерминированная сложность*, т. е. минимальное количество состояний, на которых можно построить недетерминированный конечный автомат, распознающий рассматриваемый язык. Впервые недетерминированная сложность регулярных языков рассматривалась, по-видимому, в [1]; из недавних публикаций в этой активно развивающейся области упомянем [2]–[4].

Поступила 25.05.2008

Работа выполнена при поддержке программы Рособразования “Развитие научного потенциала высшей школы”, проект 2.1.1/3537.

Как известно, класс регулярных языков замкнут относительно многих естественных операций. Это позволяет изучать не только недетерминированную сложность регулярных языков, но и недерминированную сложность операций над регулярными языками. Если операция сохраняет регулярность языка, то представляет интерес вопрос о том, какое изменение происходит в недетерминированной сложности языка при применении к нему этой операции. В [5] этот вопрос решается для основных операций над регулярными языками (объединение, пересечение, дополнение, катенация, итерация и реверсаль). В данной работе дается ответ на вопрос о недетерминированной сложности класса операций, задаваемых конечными трансдьюсерами.

В первой части работы даются необходимые определения и вспомогательные утверждения. Во второй части работы приводится доказательство основного результата — теоремы о том, что недетерминированная сложность образа регулярного языка при применении нормализованного конечного трансдьюсера не превосходит произведения числа состояний этого трансдьюсера и недетерминированной сложности исходного языка, при этом данная оценка является точной.

1. Предварительные сведения и формулировка основного результата

Напомним, что *недетерминированный конечный автомат* (НКА) — это пятерка $(Q, \Sigma, \delta, q_0, F)$, где Q — конечное множество состояний, Σ — конечное множество входных символов, $\delta \subset Q \times \Sigma \times Q$ — множество переходов, $q_0 \in Q$ — начальное состояние, $F \subset Q$ — множество заключительных состояний.

Само множество Σ, как обычно, называется *алфавитом*, его элементы — *буквами*, а конечные последовательности букв — *словами*. *Длиной* слова w называется количество букв в w и обозначается через $|w|$. *Пустое слово* — это слово нулевой длины (обозначим его через λ). Множество всех слов над алфавитом Σ обозначается через Σ^*.

Будем писать $p \xrightarrow{a} q$, если $(p, a, q) \in \delta$, и $p \overset{w}{\twoheadrightarrow} q$, если найдутся такие $a_1, \ldots, a_n \in \Sigma$ и $p_0, \ldots, p_n \in Q$, что $p = p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} p_n = q$ и $w = a_1 a_2 \ldots a_n$.

Слово w *принимается* НКА $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, если найдется такое заключительное состояние $q \in F$, что $q_0 \overset{w}{\twoheadrightarrow} q$. Языком, *распознаваемым* НКА $\mathcal{A}$, называется множество

$$L(\mathcal{A}) = \{w \in \Sigma^* \mid \exists q \in F : q_0 \overset{w}{\twoheadrightarrow} q\},$$

т. е. множество всех слов, принимаемых этим НКА.

В дальнейшем понадобится известная лемма о накачке для недетерминированных автоматов.

Лемма 1. *Пусть НКА $\mathcal{A}$ принимает слово w и длина w не меньше количества состояний автомата A. Тогда w можно представить в виде xyz, причем $|y| > 0$ и для любого $k \geq 0$ слово xy^kz принимается автоматом $\mathcal{A}$.*

По аналогии с НКА можно определить автоматы, в которых переходы могут осуществляться не только по букве из алфавита, но и по пустому слову λ. Таким образом, λ-НКА — это пятерка $(Q, \Sigma, \delta, q_0, F)$, где Q, Σ, q_0 и F определены так же, как и в обычных НКА, а множество переходов δ является подмножеством множества $Q \times (\Sigma \cup \{\lambda\}) \times Q$. В определении отношения $p \overset{w}{\twoheadrightarrow} q$ соответствующим образом меняется выбор элементов a_i: не только из алфавита Σ, но и из $\{\lambda\}$. Определения принимаемого слова и распознаваемого языка через отношение $\overset{w}{\twoheadrightarrow}$ остаются прежними.

Нетрудно доказывается

Лемма 2 ([6], теорема 2.3). *Для каждого λ-НКА существует обычный НКА, распознающий тот же самый язык и обладающий таким же количеством состояний.*

Более подробное введение в теорию формальных языков и конечных автоматов имеется в [6], [7].

Каждому регулярному языку ставится в соответствие минимально необходимое количество состояний, на которых можно построить НКА, распознающий этот язык. Может существовать несколько различных НКА с минимально возможным количеством состояний, но само это количество для данного регулярного языка определяется однозначно. Будем называть эту величину *недетерминированной сложностью языка* L и обозначать через $\mathrm{nsc}(L)$ (от *nondeterministic state complexity*). Приведем простой иллюстрирующий

Пример 1. Для $n > 0$ язык $L_n = \{a^k \mid k \equiv n-1 (\operatorname{mod} n)\}$ над алфавитом $\{a, b\}$ имеет недетерминированную сложность, равную n.

Доказательство. Язык L_n распознается НКА, изображенным на рис. 1. Этот НКА имеет n состояний, откуда $\mathrm{nsc}(L_n) \leq n$.

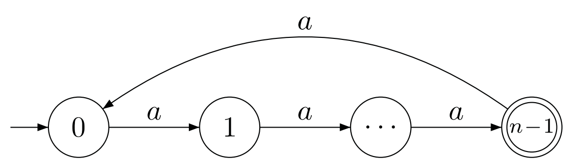

Рис. 1. НКА, распознающий язык L_n. Здесь и далее входная стрелка без подписи обозначает начальное состояние, а двойная окружность — заключительное состояние

Предположим, что язык L_n можно распознать НКА A с числом состояний меньше n. Рассмотрим слово $w = a^{n-1}$ из языка L_n. Длина w равна $n-1$, т. е. совпадает с количеством состояний НКА A. Значит, по лемме 1 из слова w можно удалить некоторую часть так, что получившееся слово тоже будет приниматься НКА A. Однако этого не может быть, так как w — слово наименьшей длины из языка L_n. Значит, язык L_n нельзя распознать НКА с числом состояний меньше n. Поэтому $\mathrm{nsc}(L_n) = n$. $\square$

Наряду с обычными недетерминированными конечными автоматами можно рассматривать недетерминированные конечные автоматы с выходом — автоматы, которые по мере чтения входной последовательности символов формируют некоторую выходную последовательность (возможно, уже над другим алфавитом). Из-за того, что автоматы с выходом удобно использовать как средство для преобразования языков, такие автоматы принято называть конечными трансдьюсерами и использовать для их обозначения строчные буквы греческого алфавита.

Конечным трансдьюсером (*абстрактным конечным автоматом* [8]) называется шестерка $(Q, \Sigma, \Delta, \delta, q_0, F)$, где Q — конечное множество состояний, Σ — конечное множество входных символов, Δ — конечное множество выходных символов, $\delta \subset Q \times \Sigma^* \times \Delta^* \times Q$ — конечное множество переходов, $q_0 \in Q$ — начальное состояние, $F \subset Q$ — множество заключительных состояний.

Для конечного трансдьюсера τ через $|\tau|$ обозначим количество состояний в этом трансдьюсере.

Будем писать $p \xrightarrow{u|v} q$, если $(p, u, v, q) \in \delta$, и $p \overset{u|v}{\twoheadrightarrow} q$, если найдутся такие $u_1, \ldots, u_n \in \Sigma^*$, $v_1, \ldots, v_n \in \Delta^*$ и $p_0, \ldots, p_n \in Q$, что $p = p_0 \xrightarrow{a_1|b_1} p_1 \xrightarrow{a_2|b_2} \cdots \xrightarrow{a_n|b_n} p_n = q$, $u = u_1 \ldots u_n$ и $v = v_1 \ldots v_n$.

Конечный трансдьюсер $\tau = (Q, \Sigma, \Delta, \delta, q_0, F)$ определяет отношение

$$\mathcal{R}(\tau) = \{(u, v) \in \Sigma^* \times \Delta^* \mid \exists q \in F : q_0 \overset{u,v}{\twoheadrightarrow} q\}.$$

Конечные трансдьюсеры τ и χ называются *эквивалентными*, если отношения $\mathcal{R}(\tau)$ и $\mathcal{R}(\chi)$ совпадают.

Также при помощи τ можно преобразовывать языки. Пусть $L \subset \Sigma^*$, тогда

$$\tau(L) = \{v \in \Delta^* \mid \exists u \in L : (u, v) \in \mathcal{R}(\tau)\}.$$

Важное значение для теории конечных трансдьюсеров имеет

Теорема 1 ([9], теорема 1.6)**.** *Для каждого конечного трансдьюсера τ и каждого регулярного языка L язык $\tau(L)$ регулярен.*

Эту теорему можно применять для доказательства того, что некоторая операция над языком сохраняет регулярность. Так, например, известно [10], что окрестность регулярного языка в метрике Хэмминга (т. е. множество слов, каждое из которых отличается от некоторого слова в исходном языке не более чем в фиксированном количестве позиций) остается регулярным языком. Получим этот результат, построив соответствующий конечный трансдьюсер.

Пример 2. Операция взятия окрестности в метрике Хэмминга сохраняет регулярность.

Доказательство. При $r \geq 0$ рассмотрим конечный трансдьюсер τ_r (рис. 2), для которого язык $\tau_r(L)$ совпадает с r-окрестностью языка L в метрике Хэмминга. Следовательно, по теореме 1 взятие окрестности некоторого радиуса в метрике Хэмминга сохраняет регулярность.

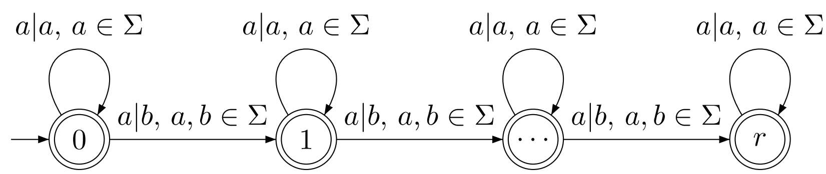

Рис. 2. Трансдьюсер τ_r

□

В приведенном примере множество переходов трансдьюсера τ_r является подмножеством множества $Q \times (\Sigma \cup \{\lambda\}) \times (\Delta \cup \{\lambda\}) \times Q$.

Конечные трансдьюсеры, обладающие таким свойством, называются *нормализованными*. Условие нормализованности не уменьшает возможности конечного трансдьюсера относительно преобразования языков. Справедлива

Лемма 3 ([11], теорема 2.24)**.** *Для любого конечного трансдьюсера существует эквивалентный ему нормализованный конечный трансдьюсер.*

Поэтому при исследовании преобразований языков с помощью трансдьюсеров можно ограничиться рассмотрением нормализованного случая. В рамках задачи о недетерминированной сложности таких преобразований это ограничение естественно еще и потому, что для произвольного трансдьюсера τ оценка сложности языка $\tau(L)$ вообще говоря не может быть дана в терминах сложности языка L и числа состояний трансдьюсера τ. Рассмотрим

Пример 3. Для $k > 0$ и конечного трансдьюсера

$$\chi_k = (\{0\}, \{a\}, \{a\}, (0, a, a^k, 0), 0, \{0\})$$

недетерминированная сложность языка $\chi_k(\{a\})$ равна $k + 1$.

Доказательство. Очевидно, трансдьюсер χ_k (рис. 3) вместо каждого вхождения в слово буквы a делает k вхождений. Следовательно, $\chi_k(\{a\}) = \{a^k\}$. Также очевидно, недетерминированная сложность языка $\{a^k\}$ равна $k + 1$. Таким образом, недетерминированная сложность применения трансдьюсера χ_k не выражается через сложность языка $\{a\}$ и количество состояний трансдьюсера χ_k.

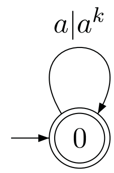

Рис. 3. Трансдьюсер χ_k

□

Основным результатом является

Теорема 2. *Пусть L — регулярный язык, τ — нормализованный конечный трансдьюсер. Тогда* $\mathrm{nsc}(\tau(L)) \leq |\tau| \cdot \mathrm{nsc}(L)$. *Эта оценка является точной, т. е. для любых $r > 1$ и $n > r + 1$ найдутся такой регулярный язык L и нормализованный конечный трансдьюсер τ, что* $\mathrm{nsc}(L) = n$, $|\tau| = r$ *и* $\mathrm{nsc}(\tau(L)) = |\tau| \cdot \mathrm{nsc}(L)$.

2. Доказательство основного результата

Сначала получим верхнюю оценку недетерминированной сложности применения нормализованного конечного трансдьюсера. Для обоснования точности верхней оценки построим, использовав примеры 1 и 2, серию конечных трансдьюсеров и регулярных языков, на которых эта оценка достигается.

Лемма 4. *Пусть L — регулярный язык, τ — нормализованный конечный трансдьюсер. Тогда*

$$\mathrm{nsc}(\tau(L)) \leq |\tau| \cdot \mathrm{nsc}(L).$$

Доказательство. Пусть $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ — НКА с минимально возможным количеством состояний, распознающий регулярный язык L. Расширим $\mathcal{A}$ до λ-НКА, добавив в каждое состояние петлю по λ, т. е. все возможные переходы вида $q \xrightarrow{\lambda} q$, где $q \in Q$.

Пусть $\tau = (P, \Sigma, \Delta, \gamma, p_0, E)$ — нормализованный конечный трансдьюсер. Можно считать, что $Q \cap P = \emptyset$.

Сначала неформально опишем основную идею доказательства. Для того чтобы построить автомат, распознающий язык $\tau(L)$, воспользуемся известной конструкцией декартового произведения автоматов. Каждое состояние нового автомата будет соответствовать паре: состояние исходного автомата $\mathcal{A}$ и состояние трансдьюсера τ. Множество переходов нового автомата будет устроено таким образом, чтобы одновременно учитывать переходы внутри исходного автомата и трансдьюсера. Чтение слова в таком автомате будет соответствовать одновременному чтению слова-прообраза в исходном автомате и чтению пары слово-прообраз и слово-образ в трансдьюсере. За счет этого построенный автомат будет распознавать язык $\tau(L)$.

На рис. 4 приводится иллюстрация подобного построения для некоторого автомата и конечного трансдьюсера τ_1 из примера 2.

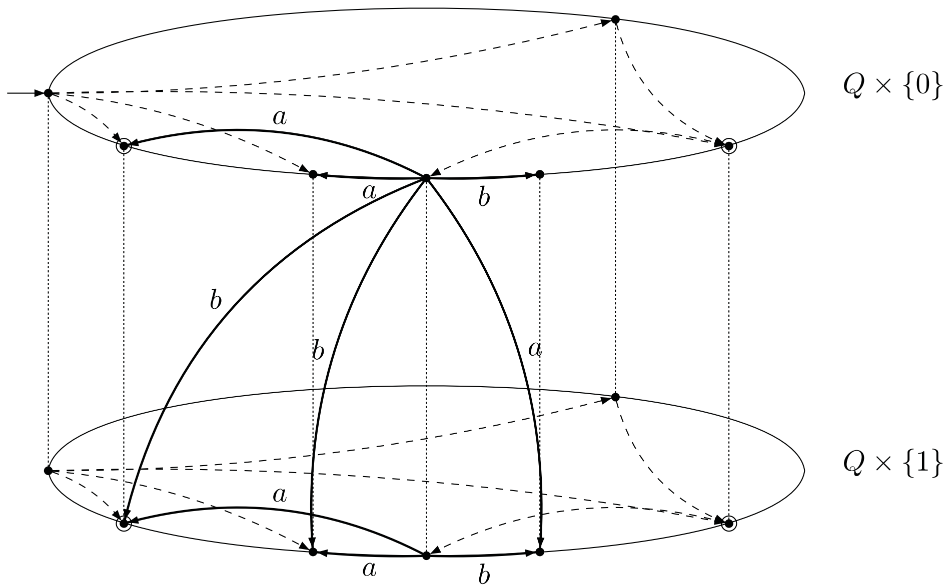

Рис. 4. Схема автомата $\mathcal{B}$ языка $\tau_1(L)$ для трансдьюсера τ_1 из примера 2

Дадим теперь формальное построение описанной конструкции. Рассмотрим λ-НКА

$$\mathcal{B} = (Q \times P, \Delta, \varepsilon, (q_0, p_0), F \times E),$$

где

$$\varepsilon = \left\{ \big((q,p), b, (q',p')\big) \mid \exists a \in \Sigma \cup \{\lambda\} : p \xrightarrow{a|b} p' \ \text{ и } \ q \xrightarrow{a} q' \right\}$$

для $q, q' \in Q$, $p, p' \in P$, $b \in \Delta \cup \{\lambda\}$.

Покажем, что $\tau(L) = L(\mathcal{B})$.

Если $v \in \tau(L)$, то существует такое слово $u \in L$, что $(u, v) \in \mathcal{R}(\tau)$. Иначе найдутся такие слова $a_1, \ldots, a_n \in \Sigma \cup \{\lambda\}$, $b_1, \ldots, b_n \in \Delta \cup \{\lambda\}$ и состояния $p_1, \ldots, p_n \in P$, что в трансдьюсере τ определены переходы $p_0 \xrightarrow{a_1|b_1} p_1 \xrightarrow{a_2|b_2} \cdots \xrightarrow{a_n|b_n} p_n$, при этом состояние p_n является заключительным, а последовательности $a_1 \ldots a_n$ и $b_1 \ldots b_n$ образуют слова u и v соответственно.

Так как u принадлежит языку L, то оно принимается автоматом $\mathcal{A}$. Это означает, что найдутся такие состояния $q_1, \ldots, q_n \in Q$, что $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \ldots \xrightarrow{a_n} q_n \in F$. Тогда слово $v = b_1 \ldots b_n$ принимается автоматом $\mathcal{B}$, поскольку имеется последовательность переходов $(q_0, p_0) \xrightarrow{b_1} (q_1, p_1) \xrightarrow{b_2} \cdots \xrightarrow{b_n} (q_n, p_n) \in F \times E$.

Пусть теперь слово v принадлежит языку $L(\mathcal{B})$, т. е. найдутся такие слова $b_1, \ldots, b_n \in \Delta \cup \{\lambda\}$, состояния $p_1, \ldots, p_n \in P$ и $q_1, \ldots, q_n \in Q$, что определена последовательность переходов $(q_0, p_0) \xrightarrow{b_1} (q_1, p_1) \xrightarrow{b_2} \cdots \xrightarrow{b_n} (q_n, p_n) \in F \times E$. Тогда по определению ε найдутся такие слова $a_i \in \Sigma \cup \{\lambda\}$, что определены переходы $p_{i-1} \xrightarrow{a_i|b_i} p_i$ и $q_{i-1} \xrightarrow{a_i} q_i$ для $i = 1, \ldots, n$. Но тогда слово $u = a_1 \ldots a_n$ принимается автоматом $\mathcal{A}$, поскольку имеется последовательность переходов $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} q_n \in F$. Значит, слово $v = b_1 \ldots b_n$ принадлежит $\tau(L)$, так как существует последовательность переходов $p_0 \xrightarrow{a_1|b_1} p_1 \xrightarrow{a_2|b_2} \cdots \xrightarrow{a_n|b_n} p_n \in E$.

Итак, язык $\tau(L)$ можно распознать при помощи λ-НКА $\mathcal{B}$. По лемме 2 можно построить обычный НКА, распознающий $\tau(L)$, с таким же, как у $\mathcal{B}$, количеством состояний. Подсчитаем количество состояний в автомате $\mathcal{B}$:

$$|Q \times P| = |Q| \cdot |P| = \mathrm{nsc}(L) \cdot |\tau|.$$

Значит, $\mathrm{nsc}(\tau(L)) \leq |\tau| \cdot \mathrm{nsc}(L)$. □

Для доказательства точности полученной оценки воспользуемся ранее построенными языком L_n из примера 1 и конечным трансдьюсером τ_r из примера 2.

Лемма 5. *Пусть $r > 0$, $n > r$. Тогда*

$$\mathrm{nsc}(\tau_r(L_n)) = |\tau_r| \cdot \mathrm{nsc}(L_n).$$

Доказательство. В соответствии с леммой 4

$$\mathrm{nsc}(\tau_r(L_n)) \leq |\tau_r| \cdot \mathrm{nsc}(L_n).$$

Предположим, что язык $\tau_r(L_n)$ можно распознать при помощи НКА, содержащего меньше $|\tau_r| \cdot \mathrm{nsc}(L_n) = (r+1)n$ состояний. Обозначим такой автомат через $\mathcal{A} = (Q, \{a, b\}, \delta, q_0, F)$.

Через $\#_a(w)$ будем обозначать количество вхождений буквы a в слово w.

Заметим, что слово w принадлежит языку $\tau_r(L_n)$ (т. е. содержится в r-окрестности языка L_n в метрике Хэмминга) тогда и только тогда, когда

$$\#_b(w) \leq r, \qquad |w| \equiv n - 1 \ (\mathrm{mod}\, n). \tag{1}$$

Рассмотрим слово $w = a^{n-1}(ba^{n-1})^r$. Поскольку оно содержит ровно r вхождений буквы b и $|w| = n-1+nr = (r+1)n-1 \equiv n-1(\mathrm{mod}\, n)$, то слово w содержится в языке $\tau_r(L_n)$, т. е. принимается автоматом $\mathcal{A}$. Так как $|w| = (r+1)n - 1$, а $|Q| < (r+1)n$, то в соответствии с леммой 1 слово w можно представить в виде xyz, где $|y| > 0$ и автомат $\mathcal{A}$ вместе со словом xyz принимает слова xy^kz для $k \geq 0$. Возможны следующие случаи.

Случай 1: слово y содержит букву b. В этом случае слово $xy^{r+1}z$ содержит не менее $r+1$ вхождения буквы b и в то же время принадлежит языку $\tau_r(L_n)$, что противоречит характеризации (1).

Случай 2: слово y не содержит букв b, а целиком состоит из букв a. Поскольку y является подстрокой слова w (которое не содержит более $n-1$ вхождения буквы a подряд), то $1 \leq |y| \leq n-1$. Слово xy^2z принадлежит языку $\tau_r(L_n)$ и

$$|xy^2z| = |xyz| + |y| \equiv n - 1 + |y| \not\equiv n - 1(\mathrm{mod}\, n),$$

что также противоречит характеризации (1).

Таким образом, предположение о том, что язык $\tau_r(L_n)$ можно распознать при помощи НКА, содержащего меньше $|\tau_r| \cdot \mathrm{nsc}(L_n)$ состояний, приводит к противоречию. Значит, $\mathrm{nsc}(\tau_r(L_n)) = |\tau_r| \cdot \mathrm{nsc}(L_n)$. □

Оценка недетерминированной сложности применения трансдьюсера τ_r была дана в работе автора [12] (в терминах взятия окрестности в метрике Хэмминга), однако эта работа содержит неточность в построении примера, на котором достигается верхняя оценка исследуемой величины.

Детерминированной сложностью регулярного языка (принято обозначать $\mathrm{sc}(L)$, от *state complexity*) называется сложность относительно количества состояний минимального детерминированного конечного автомата, распознающего этот язык. После изучения недетерминированной сложности применения конечного трансдьюсера к регулярному языку справедлив вопрос о детерминированной сложности этой операции. Лемма 4 позволяет легко получить верхнюю оценку для этой величины.

Следствие. Пусть L — регулярный язык, τ — нормализованный конечный трансдьюсер. Тогда

$$\mathrm{sc}(\tau(L)) \leq 2^{\mathrm{sc}(L)|\tau|}.$$

Доказательство. В соответствии с леммой 4 язык $\tau(L)$ можно распознать при помощи НКА, имеющего не более $\mathrm{nsc}(L)|\tau| le\, \mathrm{sc}(L)|\tau|$ состояний. Использовав стандартную конструкцию подмножеств, преобразуем полученный недетерминированный автомат в детерминированный, который будет иметь не более $2^{\mathrm{sc}(L)|\tau|}$ состояний. □

Экспоненциальный рост детерминированной сложности языка при применении конечного трансдьюсера может быть объяснен недетерминированностью, заложенной в самом трансдьюсере. Как известно, если недерминированный конечный автомат имеет n состояний, то минимальный детерминированный конечный автомат, распознающий тот же язык, может иметь 2^n состояний. Поэтому процесс детерминизации языка $\tau(L)$ может существенно увеличить количество состояний, необходимых для распознавания этого языка при помощи детерминированного конечного автомата. Приведенная экспоненциальная оценка, полученная из общих соображений, не является слишком завышенной. В работе [12] для каждого $n > 4$ строится регулярный язык, детерминированная сложность которого равна n, при этом 1-окрестности этого языка в метрике Хэмминга имеют детерминированную сложность $\frac{3}{8}n \cdot 2^n - 2^{n-4} + n$. Так как (пример 2) 1-окрестность в метрике Хэмминга можно представить в виде конечного нормализованного трансдьюсера с двумя состояниями, то верен порядок приведенной верхней оценки (по крайней мере для трансдьюсеров с двумя состояниями).

Дальнейшее сближение верхней и нижней оценок, а также построение соответствующих серий примеров для трансдьюсеров большего размера является логичным продолжением исследований в этой области.

Литература

[1] Birget J.-C. *Intersection and union of regular languages and state complexity* // Inform. Proc. Lett. – 1992. – V. 43. – № 4. – P. 185–190.

[2] Gruber H., Holzer M. *Finding lower bounds for nondeterministic state complexity is hard* // Lect. Notes Comput. Sci. – Berlin: Springer, 2006. – V. 4036. – P. 363–374.

[3] Kutrib M., Holzer M. *Unary language operations and their nondeterministic state complexity* // Lect. Notes Comput. Sci. – Berlin: Springer, 2003. – V. 2450. – P. 162–172.

[4] Salomaa K. *Descriptional complexity of nondeterministic finite automata* // Lect. Notes Comput. Sci. – Berlin: Springer, 2007. – V. 4588. – P. 31–35.

[5] Kutrib M., Holzer M. *State complexity of basic operations on nondeterministic finite automata* // Lect. Notes Comput. Sci. – Berlin: Springer, 2003. – V. 2608. – P. 148–157.

[6] Yu S. *Regular languages* // Handbook of Formal Languages. V. 1. – N.-Y.: Springer, 1997. – P. 41–110.

[7] Hopcroft J.E., Motwani R., Ullman J.D. *Introduction to automata theory, languages and computation* – MA: Addison-Wesley, 2001. – 521 p.

[8] Кудрявцев В.Б., Алешин С.В., Подколзин А.С. *Введение в теорию автоматов*. – М.: Наука, 1985. – 320 с.
[9] Sakarovitch J. *Éléments de théorie des automates*. – Paris: Vuibert Informatique, 2003. – 816 p.
[10] Calude C., Salomaa K., Yu S. *Metric lexical analysis* // Lect. Notes Comput. Sci. – Berlin: Springer, 2001. – V. 2214. – P. 48–59.
[11] Karhumäki J. *Automata and formal languages* // http://www.math.utu.fi/en/home/karhumak/Automata2007.pdf, 2007.
[12] Povarov G. *Descriptive complexity of the Hamming neighborhood of a regular language* // Proc. of Language and Automata Theory and Applications Conference. – Tarragona: Universitat Rovira i Virgili, 2007. – P. 509–520.

Г.А. Поваров

аспирант, кафедра алгебры и дискретной математики,
Уральский государственный университет,
пр. Ленина, д. 51, К-83, г. Екатеринбург, 620083,

e-mail: grigoriy.povarov@gmail.com

G.A. Povarov

Postgraduate, Chair of Algebra and Discrete Mathematics,
Ural State University,
51 Lenin str., K-83 Ekaterinburg, 620083 Russia,

e-mail: grigoriy.povarov@gmail.com