Метод распознавания множества слов через синтез детерминированного автомата

Д. В. Пархоменко

В статье предложен метод распознавания с помощью построения распознающего автомата по множеству его выходных слов. Изучены свойства «множеств с кратностями», возникающих на выходе детерминированных автоматов. Даны оценки сложности алгоритма построения автомата по мультимножеству.

Ключевые слова: автомат, детерминированный автомат, распознавание образов, выходное множество автомата, распознавание через синтез.

1. Введение

В задачах распознавания образов широко используется аппарат вероятностных источников [2, 3, 4]. Распознавание слова заключается в поиске вероятностного источника, выдающего это слово с наибольшей вероятностью. Например, в распознавании речи для каждой группы звуков строится свой вероятностный источник. Он моделирует определенный звук, и отвечает за распознавание подобных ему звуков. Несмотря на широкую популярность в приложениях, аппарат вероятностных источников имеет ряд недостатков: отсутствие правил остановки обучения, заранее неизвестное число состояний источника, а также процесс распознавания вычислительно сложен. В работе предлагается рассмотреть похожий подход, основанный на детерминированном источнике [1].

Мультимножеством называется множество, в котором допустимы вхождения нескольких копий одного и того же элемента. Мы

будем рассматривать мультимножества слов длины N в алфавите $E_k = \{0, 1, \ldots, k\}$. Под мощностью мультимножества понимается число входящих в него элементов с учетом кратностей. Мощность мультимножества далее обозначается как $|\cdot|$.

Пусть натуральное число m — порог распознавания. Через $L = (E_k, Q, E_k, \varphi, \psi, q_0)$ обозначим конечный инициальный автомат [1] с множеством состояний Q, автоматная функция которого рассматривается на словах длины N:

$$f_L : E_k^N \to E_k^N.$$

Распознавание автоматом слова $\beta \in E_k^N$ заключается в проверке, верно ли, что число прообразов $\beta \in E_k^N$ при автоматном отображении f_L не менее порога m.

$$|\{\alpha \in E_k^N : \ f_L(\alpha) = \beta\}| \geqslant m.$$

Через B_m обозначим множество слов, имеющих не менее чем m прообразов при отображении f_L. Будем говорить, что автомат L распознает множество B_m.

Построение автомата L по константе m и наперед заданному множеству B_m, назовем синтезом автомата по множеству слов B_m. Вообще говоря, в качестве распознаваемого множества можно рассматривать произвольное мультимножество B, которое должно стать образом искомой автоматной функции f_L (или войти в образ f_L):

$$f_L(E_k^N) = B, \ \ (B \subseteq f_L(E_k^N)).$$

Однако, не всякое мультимножество B может быть автоматным образом E_k^N.

Критерий того, что B является автоматным образом E_k^N, описан в теореме 1, которая дает также способ построения искомого автомата f_L. Если не представляется возможным использовать все множество $|B| = k^N$, то можно попытаться по некоторому его подмножеству $B_1 \subseteq B$ построить автомат L такой, что $B_1 \subseteq f_L(E_k^N)$. Критерий того что B_1 вложимо в автоматный образ E_k^N и алгоритм построения такого автомата описаны в теореме 4.

Будем рассматривать синтез автомата по его наперед заданному мультимножеству выходных слов, а также распознавание произвольного выходного слова этим автоматом.

2. Основные теоремы

Мультимножество B запишем в виде

$$B = \{\beta_1, \beta_1, \ldots, \beta_1, \beta_2, \beta_2, \ldots, \beta_2, \ldots, \beta_{k^N}, \beta_{k^N}, \ldots, \beta_{k^N}\},$$

где слово β_i входит во множество столько раз, какова его кратность. Через x_i обозначим кратность вхождения слова β_i. При таком представлении возможно $x_i = 0$ для некоторых $1 \leqslant i \leqslant k^N$. Считаем, что $\sum\limits_{i=1}^{k^N} x_i = k^N$. Обозначим через $\hat{B}$ множество всех собственных префиксов слов из B. Очевидно, $\hat{B}$ — также является мультимножеством, где каждому слову приписана его кратность вхождения. Через B^i обозначим подмножество $\hat{B}$, состоящее из слов длины $(N-i)$. Заметим, что сумма кратностей элементов множества B^i равна k^N для любого i. Далее везде, будем считать, что слова из множества B упорядочены лексикографически и использовать знак $\leqslant$ для лексикографического порядка. Также, введем обозначение $B' = B^1$.

Следующее утверждение содержит критерий возможности построения автомата по мультимножеству его выходных слов B.

Теорема 1. *Для существования автоматной функции f, такой что $f(E_k^N) \equiv B$ необходимо и достаточно, чтобы для любого натурального $0 < i < N$ и любого слова из $\hat{B}^i$, его кратность делилась бы на k^i.*

Следствие 1. *Существует алгоритм $\mathbf{A}_1$, который по заданному мультимножеству B находит автомат L с автоматной функцией f_L, такой что $f_L(E_k^N) \equiv B$, или определяет, что такого автомата не существует.*

Пусть мультимножество

$$B = \{\beta_1, \beta_1, \ldots, \beta_1, \beta_2, \beta_2, \ldots, \beta_2, \ldots, \beta_{k^N}, \beta_{k^N}, \ldots, \beta_{k^N}\} = \\ = \{\beta_1(x_1), \beta_2(x_2), \ldots, \beta_{k^N}(x_{k^N})\},$$

где некоторые кратности могут быть нулевыми, тогда для произвольного натурального m через mB обозначим мультимножество

$$mB = \{\beta_1(m \cdot x_1), \beta_2(m \cdot x_2), \ldots, \beta_{k^N}(m \cdot x_{k^N})\}.$$

Пусть $E_k(N)$ — множество всех слов длины не превосходящей N.

$$E_k(N) = \bigcup_{n=1}^{N} E_k^n, \;\; |E_k(N)| = \frac{k^{n+1}-1}{k-1} - 1.$$

Также для натурального $i \leqslant N$, через $B|_i$ обозначим подмножество всех слов длины i из множества B. Очевидно, $B|_i$ также является мультимножеством. Следующая теорема обобщает теорему 1 на случай когда B — суть мультимножество слов из $E_k(N)$.

Теорема 2. *Пусть B — мультимножество слов из $E_k(N)$, $|B| = |E_k(N)|$. Тогда для существования автоматной функции f, такой что $f(E_k(N)) \equiv B$ необходимо и достаточно, чтобы для любого натурального $0 < i < N$: $(B|_{i+1})' = kB|_i$.*

Рассмотрим далее функцию $\rho_\partial : \; E_k^* \cdot E_k^* \to N$, которая каждой паре слов одинаковой длины сопоставляет разность длины слова и номера первой буквы, в которой они отличаются. Для пар слов с разной длиной считаем функцию неопределенной. Если для натурального n рассмотреть дерево глубины n, а листьям дерева естественным образом приписать слова из E_k^n, то для двух произвольных $\alpha, \beta \in E_k^n$, $\rho_\partial(\alpha, \beta)$ — есть глубина минимального полного поддерева, содержащего листья α и β. Иными словами, $\rho_\partial(\alpha, \beta)$ — суть половина длины кратчайшего пути, ведущего из α в β. Например для слов $001, 101$: $\rho_\partial(001, 101) = 3$, а для 010 и 222: $\rho_\partial(010, 222) = 3$ (см. рис. 1)

Утверждение 1. *Для любых трех лексикографически упорядоченных слов $\alpha \leqslant \gamma \leqslant \beta$ одинаковой длины $\rho_\partial(\alpha, \beta) = \max(\rho_\partial(\alpha, \gamma), \rho_\partial(\gamma, \beta))$.*

Если рассматривать все слова γ, лежащие между словами α и β (в лексикографическом порядке): $\alpha \leqslant \gamma \leqslant \beta$, то верны следствия:

Утверждение 2. *Для любых слов α, β: $\alpha \leqslant \beta$, одинаковой длины $\rho_\partial(\alpha, \beta) = \max\limits_{\alpha \leqslant \gamma \leqslant \beta} (\rho(\alpha, \gamma), \rho(\gamma, \beta))$.*

Утверждение 3. *Для каждого натурального n, ρ_∂ является расстоянием на множестве E_k^n.*

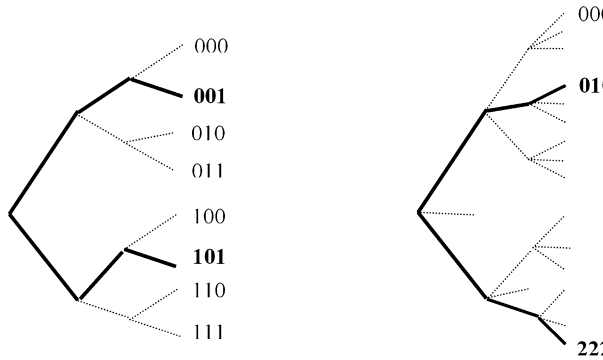

Рис. 1. Вычисление функции ρ_∂.

Определение 1. Функцию $\rho_\partial : E_k^* \times E_k^* \to \mathbb{N}$ назовем расстоянием по дереву.

Любое слово α алфавита E_k можно рассматривать как k-ичное разложение натурального числа $num(\alpha)$. Здесь и далее под «соседними» словами будем понимать слова одинаковой длины, которые, задают числа m, n: $|m - n| \leqslant 1$.

Для произвольного натурального n и произвольной словарной функции $f : E_k^n \to E_k^n$ имеет место теорема.

Теорема 3 (о сжимающем отображении).

1) *Функция f ограниченно детерминированная на словах длины n тогда и только тогда, когда для любых двух слов α, β одинаковой длины выполнено $\rho_\partial(\alpha, \beta) \geqslant \rho_\partial(f(\alpha), f(\beta))$.*
2) *Функция f ограниченно детерминированная на словах длины n тогда и только тогда, когда для любых двух соседних слов α, β из E_k^n выполнено $\rho_\partial(\alpha, \beta) \geqslant \rho_\partial(f(\alpha), f(\beta))$.*

Для произвольного упорядоченного мультимножества $B = \{\beta_1, \ldots, \beta_p\}$ слов одинаковой длины (некоторые слова могут совпадать) обозначим через

$$R(B) = \sum_{i=1}^{p-1} \rho(\beta_i, \beta_{i+1}).$$

Для произвольного упорядоченного мультимножества $B = \{\beta_1, \ldots, \beta_p\}$ и перестановки $\pi \in S_p$ обозначим через

$$\pi(B) = \{\beta_{\pi(1)}, \ldots, \beta_{\pi(p)}\}.$$

Оказывается, что лексикографический порядок, в некоторым смысле, минимален относительно расстояния по дереву. Имеет место

Утверждение 4. *Пусть B — лексикографически упорядоченное мультимножество слов одной длины. Тогда для любой перестановки $\pi \in S_{|B|}$ выполнено: $R(B) \leqslant R(\pi(B))$.*

Теорема 4.

1) *Существует алгоритм $\mathbf{A}_2$, который по мультимножеству $|B| \leqslant k^N$ находит автомат L с автоматной функцией f_L : $f_L(E_k^N) \supseteq B$, или определяет, что такого автомата не существует.*
2) *При этом число Q_L состояний автомата L удовлетворяет неравенству $Q_L \leqslant (N-1)|B| + 1$.*

Если рассматривать в качестве автоматного выхода B не мультимножество, а множество без кратностей, то оно, очевидно, будет удовлетворять первому условию теоремы 2. Более того, можно сформулировать полезное следствие.

Следствие 2. *Существует алгоритм $\mathbf{A}_3$, который по произвольному множеству слов $B = \{\beta_1, \beta_2, \ldots, \beta_p\}$ без кратностей находит автоматную функцию f: $f(E_k^N) \equiv B$.*

Таким образом, теорема 2 оценивает сложность синтеза автоматной функции f, а следующая теорема 5 — сложность проверки того, что число прообразов $\beta \in E_k^N$ при автоматном отображении f не менее порога. Имеет место

Теорема 5. *Число C арифметических операций для нахождения кратности слова β в выходном мультимножестве автомата L удовлетворяет $C = \mathrm{const} \cdot Q_L$.*

Заметим, что число операций, нужное для проверки частоты выдачи слова вероятностным источником, квадратично зависит от числа состояний источника. В детерминированном случае отпадает проблема выбора автомата для обучения: из соображений вычислительной простоты можно выбрать автомат с минимальным числом состояний. А также при синтезе детерминированного автомата не требуется остановка обучения извне.

Автор выражает благодарность своему научному руководителю — профессору МГУ д.ф.м.н Д. Н. Бабину за ценные комментарии и неоценимую помощь в работе над статьей.

3. Результаты моделирования и сложностные оценки

Для тестирования подхода была выбрана задача распознавания языка, на котором написан текст в кодировке ANSII. В качестве языков выбирались только те, которые крайне близки в синтаксическом смысле. Например, в русском и белорусском алфавитах все буквы, кроме двух, одинаковы. Соответственно и языки эти не отличимы анализом букв. Более того, из-за синтаксической близости они не отличимы также и анализом частот букв. Распределение частот встречаемости букв обоих алфавитов в различных текстах представлено на рис. 2.

Задача моделировалась для русского и белорусского языков. Набранные словари для обучения состояли из новостных статей, произведений писателей и сказаний. Общий объем словарей составлял около 100 000 слов. На языке MATLAB была написана библиотека функций, позволяющих обучать (синтезировать) детерминированные автоматы и функции для реализации алгоритма распознавания $\mathbf{A}_{rec}$.

Важно заметить, что весь белорусский словарь был «русифицирован», то есть буквы, не встречающиеся в русском алфавите были заменены на их русские аналоги. Таким образом алфавиты, на которых написаны тексты обоих словарей абсолютно неотличимы. Гистограммы на рис. 2 получены именно по таким словарям.

Для каждого языка по словарю рассчитывалась частота встречаемости пар букв. По полученной гистограмме синтезировался детер-

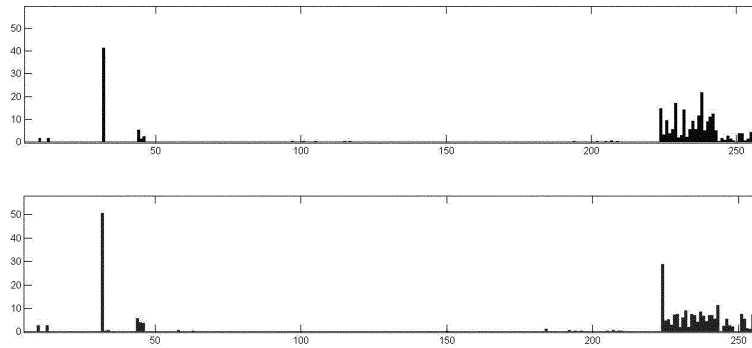

Рис. 2. Близость гистограмм русского (верхняя) и белорусского (нижняя) алфавитов.

минированный автомат. Один — для русского языка, один — для белорусского. Чтобы классифицировать входную пару букв $\alpha\beta$ применялось следующее решающее правило: определялись частоты выдачи входной пары $\alpha\beta$ для обоих автоматов, и чья частота была больше к тому языку и относилась пара. Для классификации слов (и наборов слов), слово рассматривалось как последовательность пар букв, и для каждой пары применялось решающее правило, описанной выше. Далее производилось сглаживание результата. При моделировании распознавались наборы слов длиной 39 символов. Части словарей (10 000 символов), не использовавшиеся при обучении, применялись для проверки результата. Суммарная ошибка распознавания составила 0,19.

Отметим, что пара букв в кодировке ANSII — суть 16 бит информации. Тем самым, гистограмма частот пар букв каждого языка состоит из $2^{16} = 65536$ отсчетов. Каждый отсчет может содержать 16 битное число. Следовательно, чтобы корректно сравнивать элементы двух гистограмм, а именно это и делает описанное выше решающее правило, необходимо постоянно хранить $2^{16+16+1} = 2^{33}$ бит информации. Метод распознавния через синтез автоматов позволяет существенно уменьшить использование памяти (теорема 4), незначительно увеличивая объем вычислений, нужный для классификации.

Если обозначить длину входных слов через N (считаем строго больше 1), а число состояний распознающего автомата через Q, то

для определения кратности вхождения слова в выходное множество автомата достаточно $2Q(N-1)+Q$ операций. Учитывая линейную зависимость числа состояний автомата от площади обучающей выборки (теорема 4), получаем линейную зависимость сложности распознавания от длины слова и площади обучающей выборки.

4. Доказательства

Доказательство теоремы 1. Рассмотрим представление множества слов B в виде нагруженного дерева (см. рис. 3 справа) — каждой вершине приписана кратность вхождения соответствующего слова во множество $\hat{B}$. Листьям же дерева припишем кратности соответствующих слов из B. Обозначим $T = k^N$.

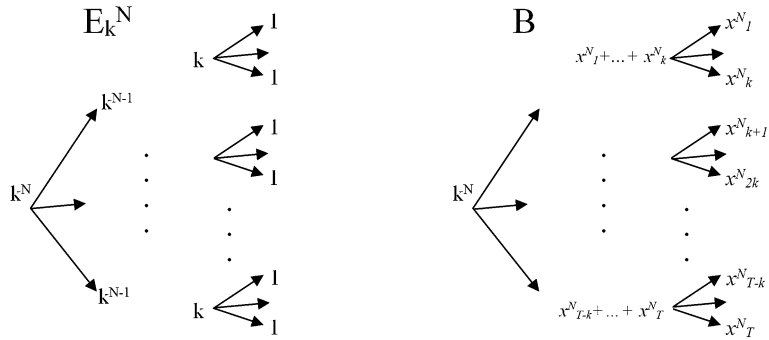

Рис. 3. Представление множества слов с кратностями в виде нагруженных деревьев.

Заметим, что некоторым листьям (вершинам) могут быть приписаны нули, так как соответствующее слово может вообще не входить в множество $B(\hat{B})$.

Пусть выполнены условия теоремы. Построим искомое автоматное отображение.

Рассмотрим N-ый уровень дерева для множества B. Будем двигаться сверху вниз (см. рис. 3 справа) по листьям деревьев, сопоставляя первому листу дерева B x_1^N первых листьев дерева E_k^N (см. рис. 3 слева), второму листу — следующие x_2^N листьев и так далее. Таким

образом листьям дерева B сопоставлены листья дерева E_k^N (причем каждый лист дерева E_k^N сопоставлен, так как сумма кратностей элементов множества B равна k^N).

Сумма кратностей, приписанных соседним (выходящим из одной вершины) листьям дерева B, делится на k по условию теоремы. Это позволяет корректно продолжить отображение листьев на вершины $N-1$-го уровня, сохраняя кратности. Вершине дерева B, из которой выходят соседние листья соспоставим вершины дерева E_k^N, из которой выходят прообразы этих листьев. Ясно, что кратности сохранятся при таком отображении (имеется в виду, что кратность вершины дерева B суть сумма кратностей сопоставленных ей вершин дерева E_k^N).

Продолжая вышеописанные шаги на следующие уровни дерева B, мы получим отображение, по построоению детерминированное и сохраняющее кратности, которое и будет задавать нам искомое автоматное отображение f.

Обратно. Пусть имеется автоматное отображение f, такое что $f(E_k^N) = B$, с учетом кратностей. Так как f детерминированное, оно задает отображение множества с кратностями $\hat{E}_k^N$ всех префиксов слов из E_k^N во множество $\hat{B}$. Кратности вхождений слов в $\hat{E}_k^N$ суть степени k (отличные от 1) или 0. Значит и кратности образов суть константы умноженные на степени k или 0. Значит, любое слово из $\hat{B}$ имеет нужную кратность. Теорема доказана.

Доказательство теоремы 2. Пусть существует автоматная функция $f : f(E_k(N)) = B$. Тогда утверждение теоремы вытекает из детерминированности f. В самом деле, пусть для всех допустимых i выполнено $(B|_{i+1})' = kB|_i$. Тогда выполнены условия теоремы 1 для множества $B|_N$. Построим отображение f : $f(E_k^N) = B|_N$. Оно и будет искомым. Действительно, легко видеть, что $f(E_k^i) = B|_i$, для всех $0 < i < N$.

Доказательство утверждения 1.

Рассмотрим произвольные α, β. Пусть $\rho_\partial(\alpha, \beta) = r$. Тогда $\alpha = \omega\delta_1\alpha_1$, $\beta = \omega\delta_2\beta_1$, где длина $|\omega| = |\alpha| - r$, а буквы δ_1 и δ_2 не совпадают. Рассмотрим произвольное γ, лежащее между словами α и β (в лексикографическом порядке): $\alpha \leqslant \gamma \leqslant \beta$. Ясно, что $\gamma = \omega\delta_3\gamma_1$, причем выполняется минимум одно из неравенств: $\delta_1 \neq \delta_3$ или $\delta_2 \neq \delta_3$.

Поэтому или $\rho_\partial(\alpha,\gamma)=r$, или $\rho_\partial(\gamma,\beta)=r$, или оба равенства верны. Следовательно, утверждение верно.

Доказательство утверждения 2. Суть, непосредственное следствие предыдущего утверждения, так как в качестве «промежуточного» слова можно выбирать любое слово, принадлежащее интервалу.

Доказательство утверждения 3. Будем рассматривать слова одной длины, тогда для введенной функции очевидно выполнены

1) $\rho_\partial(\alpha,\beta)=0 \Leftrightarrow \alpha=\beta$;
2) $\rho_\partial(\alpha,\beta)=\rho_\partial(\beta,\alpha)$;
3) $\rho_\partial(\alpha,\beta)\leqslant\rho_\partial(\alpha,\gamma)+\rho_\partial(\gamma,\beta)$, так как для $\alpha\leqslant\gamma\leqslant\beta$: $\rho_\partial(\alpha,\beta)\leqslant \max(\rho_\partial(\alpha,\gamma),\rho_\partial(\gamma,\beta))$ из утверждений 1, 2, а для γ, не принадлежащего этому интервалу или $\rho_\partial(\alpha,\beta)\leqslant\rho_\partial(\alpha,\gamma)$ или $\rho_\partial(\alpha,\beta)\leqslant\rho_\partial(\gamma,\beta)$. Следовательно для каждого натурального n, ρ_∂ — суть расстояние на множестве E_k^n.

Доказательство теоремы 3 (о сжимающем отображении).

1) Если f не о.д., то существуют два слова α,β с одинаковым началом длины n, что образы этих слов имеют одинаковое начало длины $m<n$. Значит $\rho_\partial(\alpha,\beta)<\rho_\partial(f(\alpha),f(\beta))$. Абсолютно аналогично в обратную сторону.

2) Истинность утверждения в прямую сторону следует из первого пункта.

Пусть теперь для любых α_i,α_{i+1} выполнено $\rho_\partial(\alpha_i,\alpha_{i+1})\geqslant\rho_\partial(f(\alpha_i),f(\alpha_{i+1}))$. Покажем индукцией по n, что для любых α_i,α_{i+n} верно $\rho_\partial(\alpha_i,\alpha_{i+n})\geqslant\rho_\partial(f(\alpha_i),f(\alpha_{i+n}))$. Заметим, что рассматриваются слова одной длины. База индукции ($n=1$) вытекает из формулировки. Пусть теперь утверждение верно для натурального n. Докажем для $n+1$. Обозначим попарные расстояния между словами и их образами так, как это сделано на рис. 4.

По предположению индукции верно:

$$\rho_\partial(\alpha_i,\alpha_{i+n})=r_1\geqslant r_1'=\rho_\partial(f(\alpha_i),f(\alpha_{i+n})),$$
$$\rho_\partial(\alpha_{i+n},\alpha_{i+n+1})=r_2\geqslant r_2'=\rho_\partial(f(\alpha_{i+n}),f(\alpha_{i+n+1})).$$

Нужно установить истинность:

$$\rho_\partial(\alpha_i,\alpha_{i+n+1})=r_3\geqslant r_3'=\rho_\partial(f(\alpha_i),f(\alpha_{i+n+1})).$$

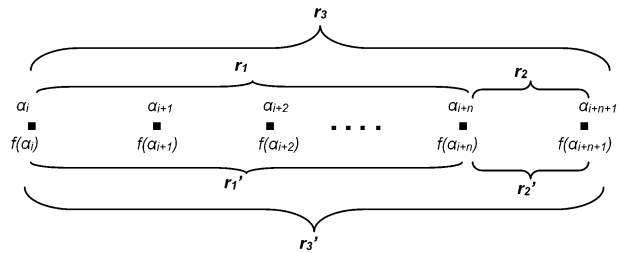

Рис. 4. Расстояния между словами и их образами.

Если $r'_1 = r'_2$, то $f(\alpha_i), f(\alpha_{i+n}), f(\alpha_{i+n+1})$ имеют общее начало длины $(N - r'_1)$, где $N = |\alpha_i|$.

Следовательно $r'_3 \leqslant N - (N - r'_1) = r'_1 \leqslant r_1 \leqslant r_3$. Последнее неравенство верно, так как слова α_i идут в лексикографическом порядке (так как они соседи).

Пусть теперь $r'_1 > r'_2$, тогда образы можно представить в виде конкатенации слов и букв:

$$\begin{gathered} f(\alpha_i) = \omega_1\delta_1\beta_1, \\ f(\alpha_{i+n}) = \omega_1\delta_2\beta_2 = \omega_2\delta'_2\beta'_2, \\ f(\alpha_{i+n+1}) = \omega_2\delta_3\beta_3, \end{gathered} \qquad (*)$$

где $|\omega_1| = (N - r'_1)$, $|\omega_2| = (N - r'_2)$, $\delta_2 \neq \delta_1$, $\delta_3 \neq \delta_2$.

Из неравенства $(N - r'_1) < (N - r'_2)$ и (*) вытекает, что $\omega_2 = \omega_1\gamma$, для некоторого слова γ. Следовательно, $f(\alpha_{i+n+1}) = \omega_1\gamma\delta_3\beta_3$. Поэтому $\rho_\partial(f(\alpha_i), f(\alpha_{i+n+1})) \leqslant (N - |\omega_1|) = r'_1 \leqslant r_1 \leqslant r_3$.

Аналогично для случая $r'_1 < r'_2$. Шаг индукции доказан. Рассмотрим теперь произвольные слова α, β[1]. Ясно, что найдутся натуральные i, n, что $\alpha = \alpha_i, \beta = \alpha_{i+n}$, следовательно $\rho_\partial(\alpha, \beta) = \rho_\partial(\alpha_i, \alpha_{i+n}) \geqslant \rho_\partial(f(\alpha_i), f(\alpha_{i+n})) = \rho_\partial(f(\alpha), f(\beta))$, что доказывает ограниченную детерминированность функции f.

Доказательство утверждения 4.

Рассмотрим произвольное мультимножество слов $B = \{\beta_1, \ldots, \beta_p\}$ одной длины. Из утверждения 3 вытекает, что для произвольных i, j выполнено: $\rho_\partial(\beta_i, \beta_j) = \max\limits_{\beta_i \leqslant \beta \leqslant \beta_j} (\rho(\beta_i, \beta), \rho(\beta, \beta_j))$.

[1]Здесь мы рассматриваем α лексикографически меньшее, чем β.

Таким образом, как бы мы не переставляли, сумма всех расстояний между соседями уменьшиться не может.

Доказательство теоремы 4.

Пусть $B = \{\beta_1(y_1), \ldots, \beta_M(y_M)\}$, где $\beta_i \in E_k^N$, $y_i \geqslant x_i$ для всех допустимых i.

Иллюстрации приведены для случая $k = 2$.

Шаг 1:

Построим дерево E_k^N глубины N, и припишем каждому листу, соответствующему слову α, деревянное расстояние между α и словом, сопоставленным вышестоящему листу (рис. 5).

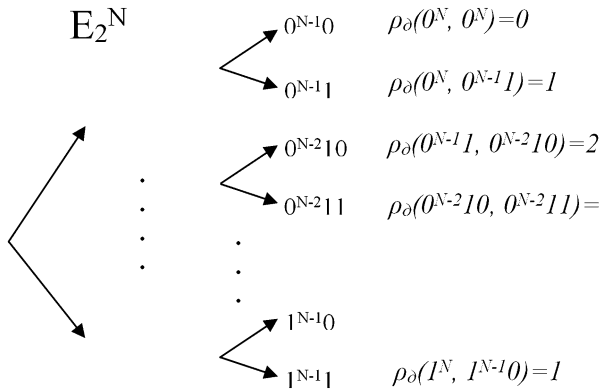

Рис. 5. Деревянное растояние между соседними листьями.

Шаг 2:

Лексикографически упорядочиваем слова множества $B = \{\beta_1(y_1), \ldots, \beta_M(y_M)\}$, $y_i \geqslant x_i$.

И сопоставляем первым x_1 листьям, начиная с листа, соответствующего 0^N (сверху вниз), слово β_1: рис. 6.

Шаг 3:

Пусть $\rho_\partial(\beta_{i-1}, \beta_i) = r_i$, $i \geqslant 2$. Будем двигаться по листья построенного дерева сверху вниз, пока не найдем лист с приписанным расстоянием $\rho_\partial(\gamma, \delta) \geqslant r_i$, где δ — приписанное этому листу слово, γ — «вышестоящее» слово.

Этому листу и $(x_i - 1)$ его соседям снизу припишем слово β_i.

Замечание 1. Если этого сделать нельзя — автомата не существует.

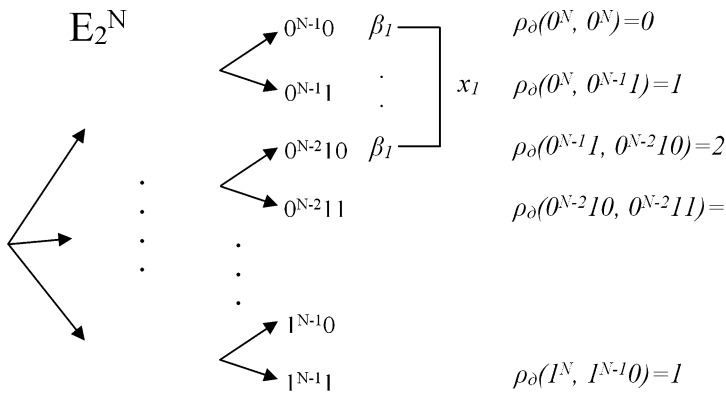

Рис. 6. Сопоставление первых x_1 слов листьям дерева.

Замечание 2. Если нет листа с условием $\rho_\partial(\gamma, \delta) \geqslant r_i$, то автомата не существует.

Шаг 4:

Повторяем Шаг 3, пока все слова множества B не окажутся сопоставленными или не обнаружится отсутствие автомата.

Шаг 5:

Если возможно, доопределим полученную функцию в тех точках, где она не была определена, словами-соседями сверху.

Пусть алгоритм $\mathbf{A}_2$ завершил работу и построил некоторую функцию f. Покажем, что f — ограниченно детерминированная. Действительно, по построению, для любых двух слов-соседей α_i, α_{i+1} выполнено $\rho_\partial(\alpha_i, \alpha_{i+1}) \geqslant \rho_\partial(f(\alpha_i), f(\alpha_{i+1}))$. Тогда по теореме о сжимающем отображении она является ограниченно детерминированной.

Покажем, что если алгоритм $\mathbf{A}_2$ обнаружил отсутствие функции f, то не существует никакой другой ограниченно детерминированной f', удовлетворяющей условиям теоремы. Для удобства изложения будем отождествлять лист дерева с приписанным ему естественным образом словом. Также будем пользоваться нумерацией листьев, возникающей при этом отождествлении[2].

[2]Лист, соответствующий числу 0 — самый первый. Далее — сверху дерева к низу.

От противного, пусть существует $f' : f'(E_k^N) = B' \supseteq B$. Рассмотрим дерево глубины N, и припишем листьям соответствующие слова — f'-образы этих листьев (слов, соответствующих листьям). Покажем, что существует алгоритм $\mathbf{B}$, переводящий это дерево в дерево — результат применения алгоритма $\mathbf{A}_2$. На 1-ом шаге алгоритма найдем минимальный в лексикографическом порядке f'-образ β_1, и сопоставим его первому листу. Пометим слово β_1. Слово, которое было сопоставленно первому листу поставим на место минимального f'-образа β_1. Далее пусть сделаны i шагов. На $(i+1)$-ом шаге выберем минимальный f'-образ β_{i+1} из всех непомеченных. Пусть $\rho_\partial(\beta_i, \beta_{i+1}) = r_i$. Сопоставим β_{i+1} тому листу, который

а) ближе всех к листу, которому сопоставлен β_i, с предыдущего шага,

б) деревянное расстояние между листами из п. а) $\geqslant r_i$.

Алгоритм останавливается, когда каждый образ становится помеченным. По построению, листьям дерева сопоставлены слова β_i в лексикографическом порядке. По утверждению 4 сумма попарных расстояний минимальна. Следовательно, каждое слово β_i будет назначено какому-либо листу, так как то же множество слов (но с не меньшей суммой расстояний между соседями) было назначено листьям при отображении f'.

По построению $\rho_\partial(\alpha_i, \alpha_{i+1}) \geqslant \rho_\partial(f(\alpha_i), f(\alpha_{i+1})) = \rho_\partial(\beta_i, \beta_{i+1})$. Теперь легко видеть, что полученное нагруженное дерево задает отображение f — результат применения $\mathbf{A}_2$. Первое утверждение теоремы доказано.

Стоит заметить, что финальный 5-й шаг в алгоритме допускает значительную свободу в выборе искомой функции. Мы можем определять неопределенные листья не из соображений соседства (словами-соседями сверху), а из других, учитывающих детерминированность результата. Например из соображений минимальности числа состояний получаемого автомата. Используя эту свободу модифицируем алгоритм для достижения оценок из второго утверждения теоремы.

Пусть для фиксированного B по алгоритму $\mathbf{A}$ построено дерево T, и автомат f с q состояниями. Ясно, что число состояний автомата q не превосходит числа всех вершин дерева T, через которые проходит путь из начальной вершины в нагруженный лист (лист, которому

приписано выходное слово). Таких вершин не более $(N-1)B+1$. Что и требовалось. Теорема доказана.

Доказательство следствия 2. Алгоритм $\mathbf{A}_3$ аналогичен алгоритму $\mathbf{A}_2$. Отличие состоит в назначении «пустых» листьев, то есть тех, которым не сопоставлено ни одно слово.

Доказательство теоремы 5. Сопоставим каждой букве b алфавита E_k квадратную целочисленную матрицу $M_b = \{m_{ij}\}$ размера $|Q|\times|Q|$, $0 \leqslant m_{ij} \leqslant N$, по правилу:

1) m_{ij} = числу способов перейти из i-го в j-ое, выдав букву b.
2) $m_{ij} = 0$, если таких способов нет.

Произвольному слову β из E_k^N сопоставим произведение матриц, сопоставленных каждой букве этого слова, причем порядок сомножителей в произведении естественным образом совпадает с порядком букв в слове β. Таким образом каждому слову из выходного алфавита сопоставлена некоторая квадратная матрица M_β. Заметим теперь, что кратность вхождения слова β в выходное множество автомата суть сумма элементов первой строки матрицы M_β. Учитывая, что для всякой буквы b в каждой строке матрицы M_b не более k ненулевых элементов, и эти матрицы фиксированы получаем утверждение теоремы.

Список литературы

[1] Кудрявцев В. Б., Алешин С. В., Подколзин А. С. Введение в теорию автоматов. М.: Наука, 1985.

[2] Бабин Д. Н., Мазуренко И. Л., Холоденко А. Б. О перспективах создания системы автоматического распознавания слитной устной русской речи // Интеллектуальные системы. Т. 8, вып. 1–4. М., 2004. С. 45–70.

[3] Пархоменко Д. В. Моделирование вероятностными источниками // Интеллектуальные системы. Т. 14, вып. 1–4. М., 2010. С. 213–223.

[4] Левинсон С. Е. Структурные методы автоматического распознавания речи // ТИИЭР № 11, ноябрь 1985. С. 100–126.