

Федеральное государственное бюджетное образовательное учреждение
высшего образования
Московский государственный университет имени М.В. Ломоносова
Механико-математический факультет
Кафедра Математической Теории Интеллектуальных Систем

На правах рукописи

Пархоменко Денис Владимирович

ГИСТОГРАММНАЯ ФУНКЦИЯ АВТОМАТА И ЕЕ ПРИЛОЖЕНИЯ

01.01.09. – дискретная математика и математическая кибернетика

диссертация на соискание ученой степени
кандидата физико-математических наук

Научный руководитель:

Доктор физико-математических наук,
профессор Д.Н. Бабин

Москва- 2015

Оглавление

Оглавление	2
Введение.....	3
Глава 1. Гистограммная функция автомата и порождаемые ею языки.....	15
1.1 Гистограммная функция автомата и ее свойства.....	15
1.2 Классы языков, порожденные гистограммной функцией автомата.	24
1.3 Критерий принадлежности языка к классу $\mathcal{L}_p$	31
1.4 Доказательство критерия принадлежности языка к классу $\mathcal{L}_p$. (теоремы 1.5, 1.6)	
.....	33
Глава 2. Автоматные мультимножества и распознавание с помощью	
гистограммной автоматной функции.	45
2.1 Алгоритм восстановления гистограммной автоматной функции по	
конечному мультимножеству.	45
2.2 О классах L_p - языков.	66
Глава 3. Распознавание через синтез.	72
3.1 Распознавание слов автоматами.	72
3.2 Пример использования гистограммной автоматной функции для	
распознавания слов.	79
Литература:.....	84
Публикации автора по теме диссертации	85

Введение.

Теория автоматов - раздел дискретной математики, возникший в середине 20-го века в связи с изучением свойств конечных автоматов.

Конечный автомат принято определять как устройство, имеющее входной, выходной канал и конечное число состояний.

Функционирование автомата осуществляется в дискретной шкале времени: в текущий момент автомат получает на вход один из конечного множества входных сигналов, изменяет текущее состояние и выдает на выход один из конечного множества выходных сигналов. Таким образом, автомат производит отображение входных последовательностей в выходные. Связанное с автоматом отображение называется автоматной функцией.

Сами автоматы и их алгебры начали исследоваться в тридцатые годы предыдущего столетия, но особенно активно в период 50-х годов. С самого возникновения понятия конечного автомата в математике исследовались свойства автоматов порождать языки. Фундаментальный результат в области описания языков, которые можно порождать автоматами опубликовал С.К.Клини [1] в 1956. Клини показал, что представимые в конечных автоматах множества слов в точности совпадают с регулярными языками.

Предпосылкой к формированию теории автоматов явилась фундаментальная статья Мак-Каллока и Питтса [14] о логическом анализе нервной деятельности. Однако, исторически первой работой, давшей толчок к развитию теории автоматов, является работа Э. Поста 1941 года [2]. В ней была описана структура решетки замкнутых классов булевых функций. Булевы функции являются частным случаем автоматов - автоматами без памяти. Возникшие для булевых функций, а также для функции k -значной логики [15], задачи о выразимости, полноте, базисах актуальны и для автоматных функций. Применим к ним и аппарат, используемый для решения этих задач. Под выразимостью здесь понимается возможность получения одной автоматной функций через другие автоматные функции. Частным случаем задачи выразимости является задача полноты, в которой проверяется возможность выразимости всех автоматных функций. Этот подход носит название структурной теории автоматов. Значительный вклад в развитие структурной теории автоматов внесли В.Б.Кудрявцев [4], А.А.Летичевский [12], М.И.Кратко [13], С.В. Алёшин [16], Д.Н.Бабин [3], В.А. Бувеч.

В настоящей работе автор предлагает новый тип поведения автоматов, когда множество распознаваемых автоматом слов порождается частотными свойствами автоматной функции.

Под конечным детерминированным инициальным автоматом V , согласно [4], будем понимать шестерку

$$V=(A,Q,B,\varphi,\psi,q_0),$$

где A - входной алфавит, Q – множество состояний, B – выходной алфавит, все три множества конечны, $q_0 \in Q$ - начальное состояние.

Функционирование автомата происходит по тактам времени t , согласно каноническим уравнениям:

$$\begin{cases} q(0) = q_0 \\ q(t+1) = \varphi(q(t), a(t)), \\ b(t) = \psi(q(t), a(t)) \end{cases}$$

где φ, ψ – суть функции переходов и выходов, соответственно.

Через A^* будем обозначать множество слов в алфавите A . Следуя С.К.Клини, будем говорить, что автомат представляет язык L с помощью множества выходных символов $B' \subseteq B$, если

$$L = \{ \alpha \in A^* \mid \psi(q_0, \alpha) \in B' \}.$$

Регулярным языком над алфавитом A , согласно [4], будем называть множество слов, которое можно получить из пустого слова и букв алфавита применением конечного числа операций объединения, конкатенации и итерации. Теорема Клини утверждает, что регулярные языки это в точности языки представимые в автоматах.

Естественным расширением концепции конечного детерминированного автомата, является понятие вероятностного автомата. Согласно Бухараеву Р.Г. [5], конечным вероятностным автоматом назовем пятерку

$$\mu = (S, E_n, E_m, v(s', a \mid s, b), v_0),$$

где $S = \{s_1, s_2, \dots, s_N\}$ - совокупность состояний автомата, n, m – натуральные числа, E_n, E_m – входной и выходной алфавиты, а числа

$$v(s', a \mid s, b)$$

удовлетворяют условиям:

$$\begin{aligned} v(s', a \mid s, b) &\geq 0, \\ \sum_{(s', a) \in S \times E_n} v(s', a \mid s, b) &= 1 \end{aligned}$$

$$\text{где } s, s' \in S, a, a' \in E_n, b \in E_m.$$

Стохастический вектор v_0 задает начальное распределение вероятностей на множестве состояний S .

При содержательном толковании вероятностного автомата, как преобразователя информации, число $v(s', a \mid s, b)$ означает условную

вероятность перехода из состояния s' при подаче входного символа a , в состояние s при выходном сигнале b . Числа

$$v(s', a | s, b)$$

иногда представляют в виде множества матриц

$$\{ A(a), B(a) \mid a \in E_m \},$$

где стохастическая матрица $A(a)$ - соответствует переходам между состояниями, а стохастическая матрица $B(a)$ – выходам, при заданной букве a входного алфавита.

Важным подклассом вероятностных автоматов являются вероятностные источники, или скрытые марковские модели (СММ). Концепция СММ была сформулирована Баумом в конце 1960-х. СММ играют важную роль в приложениях: предсказании погоды, распознавании образов, машинном обучении. Вероятностный источник [6] это пятерка

$$\lambda = (S, A, F, G, \pi)$$

где

$$S = \{s_1, s_2, \dots, s_N\}$$

- совокупность состояний источника,

$$A = \{a_1, a_2, \dots, a_M\}$$

- алфавит выходной последовательности,

$$F = \{f_{ij}\}, \text{ где } f_{ij} = P[q_{t+1} = s_j \mid q_t = s_i],$$

- стохастическая матрица вероятностей переходов между состояниями,

$$G = \{g_{ij}\}, \text{ где } g_{jk} = g_j(k) = P[a_k \mid q_t = s_j],$$

- стохастическая матрица вероятностей появления символов

$\pi = (\pi_1, \pi_2, \dots, \pi_N)$ - вектор вероятностей начального состояния

$$\pi_i = P(q_0 = s_i), \quad 1 \leq i, j, k \leq N.$$

Вероятностный источник – это вероятностный автомат без входа, который приписывает вероятности выходным словам. Выбирая параметры S, A, F, G, π подходящим образом, вероятностные источники

используют для описания и распознавания слов, имеющих разную вероятность наблюдения. Для заданной модели и порога вероятности, модель будет распознавать все те слова, которые могут появиться на ее выходе с вероятностью более некоторого порога.

Для $\varepsilon > 0$ обозначим через

$$\Gamma(\varepsilon) = \{ \gamma \in A^* \mid P(\gamma | \lambda) \geq \varepsilon \}$$

- множество выходных слов вероятностного источника λ , появляющихся на его выходе с вероятностью большей или равной ε .

В реальных задачах вероятностные источники используют для «распознавания через синтез», когда для каждой из гипотез строится свой вероятностный источник, её описывающий. Для исследуемого слова принимается та гипотеза, источник которой выдаёт на этом слове максимальную вероятность.

В данной работе предложен способ определения весовых функций на словах выходного алфавита с помощью конечного детерминированного автомата.

Пусть задан конечный детерминированный инициальный автомат

$$V = (A, Q, B, \varphi, \psi, q_0).$$

Гистограммной автоматной функцией автомата V назовем функцию

$$\kappa_V: B^* \rightarrow \aleph \cup \{0\},$$

определенную по правилу

$$\kappa_V(\beta) = |\{ \alpha \in A^* \mid \bar{\psi}(q_0, \alpha) = \beta \}|.$$

Здесь $\aleph$ означает множество натуральных чисел, а $|\cdot|$, означает мощность множества. Для любого натурального p и конечного детерминированного автомата V , p -языком, порожденным автоматом V назовем:

$$L_p(V) = \{ \beta \in B^* \mid \kappa_V(\beta) \geq p \}.$$

Автор показал, что каждый конечный детерминированный инициальный автомат V , задает цепочку регулярных, вложенных друг в друга языков:

$$L_1(V) \supseteq L_2(V) \supseteq L_3(V) \dots$$

Автором изучаются языки такого типа, получен ряд теорем об их свойствах.

Автор выражает глубокую благодарность своему учителю – доктору физико-математических наук, профессору Бабину Д.Н. за постановку задачи и внимание к работе. Автор благодарит коллектив кафедры Математической Теории Интеллектуальных Систем за доброжелательную атмосферу и ценные советы. Отдельную благодарность автор выражает сотруднику кафедры к.ф.м.н. П.А.Пантелееву.

Содержание работы.

Центральным понятием **Главы 1** является понятие гистограммной автоматной функции, замечательной тем, что с помощью нее можно ещё одним способом порождать регулярные языки конечным детерминированным автоматом. **Глава 1** посвящена изучению таких языков, а также свойств гистограммной автоматной функции.

Определение 1.1:

Пусть задан конечный детерминированный инициальный автомат

$$V=(A, Q, B, \varphi, \psi, q_0).$$

Гистограммной автоматной функцией автомата V назовем функцию

$$\kappa_V: B^* \rightarrow \aleph \cup \{0\},$$

определенную по правилу

$$\kappa_V(\beta) = |\{ \alpha \in A^* \mid \bar{\psi}(q_0, \alpha) = \beta \}|.$$

Здесь $\mathbb{N}$ означает множество натуральных чисел, а $|\cdot|$, означает мощность множества.

В параграфе 1.1 для произвольного натурального p и автомата V рассматриваются языки, состоящие из слов, встречающихся на выходе автомата V не менее p раз.

Определение 1.2:

Для любого натурального p и конечного детерминированного автомата V , p -языком, порожденным автоматом V назовем:

$$L_p(V) = \{ \beta \in B^* \mid \kappa_V(\beta) \geq p \}.$$

Каждый конечный детерминированный инициальный автомат V , задает цепочку языков, вложенных друг в друга языков:

$$L_1(V) \supseteq L_2(V) \supseteq L_3(V) \dots$$

Структуру этих языков описывают следующие теоремы:

Теорема 1.1:

Для любого конечного инициального автомата A и для всякого $i > 0$, $L_i(A)$ – регулярный язык.

Теорема 1.2:

Пусть дан автомат V , у которого мощность выходного алфавита не превосходит мощности входного. Если для некоторого $i \in \mathbb{N}$, слово $\beta \in L_i(V)$, то найдется буква b выходного алфавита автомата V такая, что $\beta b \in L_i(V)$.

Параграф 1.2 посвящен изучению свойств классов языков:

Определение 1.3:

Назовём классом p -языков множество

$$\mathcal{L}_p = \{ L_p(A) \mid A \in K(A, B) \},$$

где $K(A, B)$ – класс всех автоматов вида $(A, Q, B, \varphi, \psi, q_0)$.

Исследована замкнутость введенных классов языков относительно основных операций и их связь с регулярными, факториальными и другими классами языков.

Имеет место

Теорема 1.3:

- 1) Класс языков $\mathcal{L}$ замкнут относительно операций: объединения, взятия автоматного образа, конкатенации и итерации, но не замкнут относительно пересечения.
- 2) Класс языков $\mathcal{L}$ при $i > 1$ замкнут относительно операции взятия автоматного образа, но не замкнут относительно теоретико-множественных операций объединения и пересечения.

Каждый p -язык является регулярным, но не каждый регулярный является p -языком. Вопрос, проверяемо ли свойство произвольного языка быть языком p -типа, исследуется в **Параграфе 1.3**. Более точно в этом разделе будет сформулирован критерий принадлежности языка к классу $\mathcal{L}_p$ для произвольного p .

Для того чтобы явно представить алгоритм, будет введено понятие p -раскраски автомата без выхода.

Определение 1.4:

Пусть задан автомат

$$W = (B, Q, \varphi, q_0, Q_F)$$

без выхода, с выделенным множеством финальных состояний Q_F .

Введем функцию

$$k: Q \rightarrow \mathbb{N} \cup \{0\},$$

сопоставляющую каждому состоянию из Q неотрицательное число.

Будем говорить, что функция k задает на автомате V правильную p -раскраску состояний, если для любой вершины $q \in Q$ выполнены:

- 1) $k(q) < p$, для любого $q \notin Q_F$,
- $k(q) \geq p$, для любого $q \in Q_F$ и
- $k(q_0) = 1$.

$$2) \quad k(q) \cdot |B| = \sum_{i=1}^{|B|} k(\varphi(q, b_i)), \quad \text{если } \forall i \quad \varphi(q, b_i) \notin Q_F \quad \text{и}$$

$$k(q) \cdot |B| \geq \sum_{i, \varphi(q, b_i) \notin Q_F} k(\varphi(q, b_i)) + s(q) \cdot p, \quad \text{где } s(q) = |B| - \left| \{i \mid \varphi(q, b_i) \notin Q_F\} \right|$$

Автомат V , в таком случае назовем правильно **p**-раскрашенным.

Заметим, что не всякий автомат может быть правильно **p**-раскрашен.

Центральными результатами работы являются следующие теоремы.

Теорема 1.5:

Пусть $p > 1$, автомат $V = (B, Q, \varphi, q_0, Q_F)$ представляет язык L . Тогда $L \in \mathcal{L}_p$ в том и только том случае, когда найдется правильно **p**-раскрашенный автомат

$$V' = (B, Q', \varphi', q_0', Q_F'),$$

также, представляющий L . Причем число состояний нового автомата сравнимо с числом состояний исходного, а именно:

$$|Q'| \leq \frac{p(p-1)}{2} \cdot |Q \setminus Q_F| + p \cdot |Q_F|.$$

Теорема 1.6:

Существует алгоритм, который для любого регулярного языка L , заданного представляющим автоматом, определяет все те p , для которых $L \in \mathcal{L}_p$.

Параграф 1.4 содержит доказательства основных результатов сформулированных в **Параграфе 1.3**.

Вопрос синтеза гистограммной функции исследован в **Главе 2**. Под мультимножеством над множеством B — будем понимать пару

$$(\bar{A}, \chi),$$

где $\chi: \bar{A} \rightarrow \mathbb{N}$ — это функция, сопоставляющая каждому элементу $\bar{A}$ натуральное число, называемое кратностью этого элемента. Задача

восстановления гистограммной функции сводится к построению детерминированного автомата, гистограммная функция которого на множестве слов $\bar{A}$ совпадает с отображением χ . Заметим, что восстановить гистограммную функцию не всегда возможно. Однако, можно восстановить гистограммное накрытие. Мультимножество (B, χ) назовем автоматным, если $\kappa_V(\beta) = \chi(\beta)$, для любого $\beta \in B$, и назовём квази-автоматным, если существует автомат

$$V = (E_k, Q, E_k, \varphi, \psi, q_0),$$

что для любого слова $\beta \in B$ $\kappa_V(\beta) \geq \chi(\beta)$. Функцию κ_V , в этом случае, назовем гистограммным накрытием. В параграфе 2.1 сформулированы критерии автоматности и квази-автоматности конечного мультимножества слов и алгоритм построения автомата с заданной гистограммной функцией.

Теорема 2.3:

1) Существует алгоритм $\mathcal{A}$, который по мультимножеству (B, χ) , $\sum_{\beta \in B} \chi(\beta) \leq k^N$ находит автомат V , что κ_V - гистограммное накрытие (B, χ) , или определяет, что такого автомата не существует.

2) Число операций сложения и умножения натуральных чисел, необходимых для проверки и поиска не превосходит

$$C \cdot \sum_{\beta \in B} \chi(\beta).$$

3) При этом число $|Q_V|$ состояний автомата V удовлетворяет неравенству

$$|Q_V| \leq \sum_{\beta \in B} \chi(\beta) \cdot \frac{1 - \frac{1}{k^{M-1}}}{k-1} + M - 1,$$

где M – максимальная длина слова из B .

Для приложений важна простота подсчета значения гистограммной функции детерминированного автомата. Оказывается, гистограммную функцию можно вычислять эффективно.

Теорема 2.4:

Пусть $V=(A,Q,B,\varphi,\psi,q_0)$, k_V – его гистограммная функция. Тогда для любого слова β выходного алфавита B , число C арифметических операций для нахождения $k_V(\beta)$ удовлетворяет $C = |B||\beta|*|Q|$.*

Параграф 2.2 посвящен необычному, на первый взгляд, свойству классов языков $\mathcal{L}$. Если для некоторых $j \leq i$, $L \in \mathcal{L}_i$ будет выполнено $L \in \mathcal{L}_j$, то отсюда не следует, что для классов языков выполнено $\mathcal{L}_i \subseteq \mathcal{L}_j$. Оказывается, верна

Теорема 2.5:

Для любых натуральных $i \neq j$ выполнено: $\mathcal{L}_i \not\subseteq \mathcal{L}_j$.

В **Главе 3** описан опыт применения алгоритма синтеза гистограммного накрытия к задаче распознавания образов. **Параграф 3.1** посвящен методам распознавания слов автоматами и источниками, дано описание метода распознавания с помощью гистограммной автоматной функции. Для проверки нового метода распознавания был написан программный комплекс на ЭВМ и собрана коллекция данных ЭКГ. ЭКГ были выбраны для анализа, т.к. с одной стороны изменение потенциалов сердечной мышцы, регистрируемое электро-кардиографом обладает регулярной структурой и будет релевантно методу распознавания. С другой стороны, ЭКГ обладают достаточной вариабельностью, чтобы можно было говорить о чистоте эксперимента. Данные для распознавания и обучения были получены от Института Клинической Кардиологии им. А.Л.Мясникова. Полный корпус данных состоял из 100 ЭКГ здоровых испытуемых и 100 ЭКГ испытуемых с нарушением проводимости сердца. Задача распознавания состояла в выявлении ЭКГ

здорового человека. В **параграфе 3.2** приведены результаты симуляции метода. Доля правильно классифицированных ЭКГ превысила 0.82.

Глава 1. Гистограммная функция автомата и порождаемые ею языки.

1.1 Гистограммная функция автомата и ее свойства.

Рассмотрим конечные алфавиты A, B :

$$A = \{a_1, \dots, a_n\}, B = \{b_1, \dots, b_m\}$$

и конечный детерминированный инициальный автомат

$$V = (A, Q, B, \varphi, \psi, q_0).$$

Автомат задает ограниченно-детерминированную словарную функцию $f_V: A^* \rightarrow B^*$ по правилу

$$f_V(\alpha) = \bar{\psi}(q_0, \alpha),$$

где словарная функция $\bar{\psi}(q_0, \alpha\alpha)$ определяется по рекуррентному соотношению

$$\bar{\psi}(q_0, \alpha\alpha) = \bar{\psi}(q_0, \alpha) \psi(\varphi(q_0, \alpha), \alpha),$$

для произвольных $a \in A$, $\alpha \in A^*$. Если функция f_V является биективной, то автомат V называется обратимым.

Определение 1.1:

Пусть задан конечный детерминированный инициальный автомат V . Гистограммной автоматной функцией автомата V назовем функцию

$$\kappa_V: B^* \rightarrow \mathbb{N} \cup \{0\},$$

определенную по правилу

$$\kappa_V(\beta) = |\{ \alpha \in A^* \mid f_V(\alpha) = \beta \}|.$$

Утверждение 1.1:

Для любого конечного детерминированного инициального автомата $V=(A,Q,B,\phi,\psi,q_0)$, его гистограммная функция κ_V удовлетворяет соотношениям:

- 1) $\sum_{\beta \in B^N} \kappa_V(\beta) = |A|^N$, для любого натурального $N \in \mathbb{N}$.
- 2) Если для любого $\beta \in B^*$, $\kappa_V(\beta) = 1$, то автомат V - обратимый.
- 3) Если автомат V – обратимый, то для любого $\beta \in B^*$, $\kappa_V(\beta) = 1$.
- 4) Если автомат V – константный, то для любого натурального N найдется $\beta \in B^N$, $\kappa_V(\beta) = |A|^N$.

Доказательство:

Первый пункт непосредственно вытекает из определения функции. Мощность образа совпадает с мощностью прообраза для инъективного отображения, определенного на конечных множествах.

Так как значение $\kappa_V(\beta)$ гистограммной автоматной функции на некотором слове β суть не что иное, как число прообразов слова β при автоматном отображении f_V , то условие в п.2 означает, что у любого

слова ровно один прообраз для заданного автомата V . Это эквивалентно обратимости автомата.

Третий пункт вытекает из того, что обратимый автомат переводит любые 2 различных входных слова в 2 различных выходных.

Последний пункт утверждения – следствие того, что константный автомат переводит все входные слова одной длины в одно и то же выходное слово. ■

Языком над алфавитом B будем называть произвольное множество слов этого алфавита. Язык L алфавита B называется *автоматно перечислимым* [4], если найдется автомат $V = (A, Q, B, \varphi, \psi, q_0)$, что $L = \{ \bar{\psi}(q_0, \alpha) \mid \alpha \in A^* \}$.

Произвольный непустой язык L алфавита B назовем *продолжаемым*, если для каждого слова $\beta \in B^*$ найдется буква $b \in B$, такая что $\beta b \in B^*$. Класс всех продолжаемых языков обозначим через $Cont$, класс языков не являющихся продолжаемыми – $ACont$.

В работе [4] установлено, что язык является автоматно-перечислимым в том, и только том случае, когда он регулярный, префиксный (замкнут относительно операции взятия собственных префиксов) и продолжаемый.

Определение 1.2:

Для произвольного конечного детерминированного автомата V , положим

$$L_i(V) = \{ \beta \in B^* \mid \kappa_V(\beta) \geq i \}.$$

Справедлива

Теорема 1.1:

Для любого конечного инициального автомата V и для всякого $p \geq 0$, $L_p(V)$ – регулярный язык.

Доказательство:

Зафиксируем произвольный конечный автомат $V = (A, Q, B, \varphi, \psi, q_0)$ и натуральное p . Рассмотрим $L_p = L_p(V)$ – множество слов, встречающихся

на выходе автомата V не менее p раз. Сопоставим каждой букве b алфавита B квадратную целочисленную матрицу

$$M_b = \{m_{ij}\}$$

размера $|Q| \times |Q|$, $0 \leq m_{ij} \leq |A|$, по правилу:

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

Произвольному слову β из B^* сопоставим произведение матриц, соответствующих 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 квадратная матрица M_β . При этом (i,j) -ый элемент матрицы M_β – это число входных слов переводящих i – тое состояние в j – тое состояние, и выдающих слово β .

Пусть τ – квадратная матрица размера $|Q| \times |Q|$, элементы этой матрицы принимают значения $-1, 0, 1, \dots, p-1$. Разобьем множество неотрицательных матриц M размера $|Q| \times |Q|$ на непересекающиеся классы $\mathcal{K}_\tau = \{ M = \{m_{ij}\} \}$ где $m_{ij} > p-1$, если в матрице τ , $\tau_{ij} = -1$ и $m_{ij} = \tau_{ij}$ иначе. Так как матриц τ конечное число, то различных классов $\mathcal{K}_\tau$ также конечное число. Заметим, что любая квадратная целочисленная матрица размера $|Q| \times |Q|$ лежит в каком-то классе, причем только в одном. Класс $\mathcal{K}_\tau$ состоит из всех матриц, которые в выделенных (в матрице τ числом -1) позициях имеют значение больше чем $p-1$, а в остальных позициях значение $m_{ij} = \tau_{ij}$. Построим источник в виде дерева, представляющий язык L_p . Вершины дерева – это классы $\mathcal{K}_\tau$. Рёбро дерева с отметкой b проводится по правилу:

- 1) из $\mathcal{K}_{\tau_1}$ в $\mathcal{K}_{\tau_2}$ по b , если $\mathcal{K}_{\tau_1} * M_b \subseteq \mathcal{K}_{\tau_2}$, где M_b – матрица, соответствующая символу b алфавита B . (Здесь $*$ означает умножение матриц).

2) Корнем дерева является класс $\mathcal{K}_\tau$ для единичной матрицы $\tau = E$.

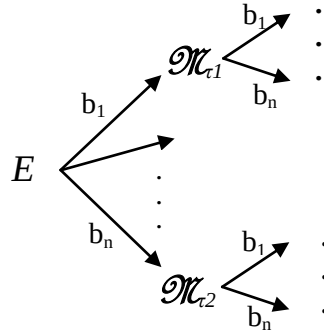


Рисунок 1. Дерево источника, представляющего L_p .

Удостоверимся, что все матрицы из класса $\mathcal{K}_{\tau_l}$ при умножении на целочисленную матрицу M с неотрицательными элементами переходят в один и тот же класс $\mathcal{K}_{\tau_2}$. Пусть M_{τ_l} минимальный элемент класса $\mathcal{K}_{\tau_l}$. Пусть $R = M_{\tau_l} * M \in \mathcal{K}_{\tau_2}$. Для любой матрицы $M_1 \in \mathcal{K}_{\tau_l}$ выполнено $M_{\tau_l} \leq M_1$, $R \leq M_1 * M = R_1$. Покажем, что матрицы R_1 и R могут отличаться только в позициях со значениями большими $(p-1)$. Пусть (i,j) -ый элемент матрицы R не равен соответствующему элементу матрицы R_1 . Это означает, что

$$\sum_{k=1}^n M_1(i, k) \cdot M(k, j) > \sum_{k=1}^n M_{\tau_l}(i, k) \cdot M(k, j)$$

(обратное невозможно т.к. $R \leq R_1$), следовательно найдется $l < n$:

$$M_1(i, l) > M_{\tau_l}(i, l) \text{ и } M(l, j) > 0.$$

Т.к. матрицы M_1 и M_{τ_l} из одного класса по построению, такое возможно лишь в случае $M_1(i, l) > M_{\tau_l}(i, l) \geq p$, что означает

$$M_1(i, l) * M(l, j) > R = M_{\tau_l}(i, l) * M(l, j) \geq p.$$

Следовательно, R_1 принадлежит тому же классу матриц, что и R , т.е. $M_1 * M \in \mathcal{K}_{\tau_2}$. Что и требовалось

Построим источник по данному дереву. Объявим финальными состояниями, которым соответствуют матрицы, с суммой элементов в первой строке большей $p-1$. Данный источник представляет язык L_p , так как сумма элементов в первой строке матрицы более $p-1$ означает, что

слово, сопоставленное этой матрице, выдается более $p-1$ раза и наоборот.



Далее мы ограничимся рассмотрением только таких автоматов, у которых мощность выходного алфавита B не превосходит мощности входного A , $|A| \geq |B|$.

Теорема 1.2:

Для любого конечного инициального автомата V , такого, что $|A| \geq |B|$ и для всякого $i \geq 0$, $L_i(V)$ – продолжаемый язык.

Доказательство:

Рассмотрим детерминированный автомат V с функцией переходов φ и выходов – ψ , входным алфавитом $A = \{a_1, \dots, a_n\}$ и выходным – $B = \{b_1, \dots, b_m\}$, $m \leq n$. Пусть $\beta \in L_i(V)$, тогда найдется i входных слов $\alpha_1, \dots, \alpha_i$, которые переводят автомат в состояния $q_1', \dots, q_i'$, соответственно, и выдают на выходе слово β . Для каждого состояния q_k' из $q_1', \dots, q_i'$ выходная функция

$$\psi(q_k', \cdot): A \rightarrow B$$

принимает не более m значений. Рассмотрим $i \times n$ матрицу M , у которой p, r -ый элемент суть выходная буква $\psi(q_p', a_r)$. Тогда существует буква выходного алфавита b , которая встречается в строке

$$\psi(q_1', a_1), \dots, \psi(q_1', a_n), \psi(q_2', a_1), \dots, \psi(q_2', a_n), \dots, \psi(q_i', a_n)$$

не менее i раз. Если бы таковой не существовало, то матрица M состояла бы не более чем из $(i-1) \cdot m$ элементов, но $m \leq n$, следовательно, $(i-1) \cdot m < i \cdot n$, что противоречиво. Данная буква b задает искомое $\beta b \in L_i(V)$. ■.

Следствие 1.1:

Для любого конечного инициального автомата V и для всякого $i \geq 0$, $L_i(V)$ – либо бесконечное либо пустое множество.

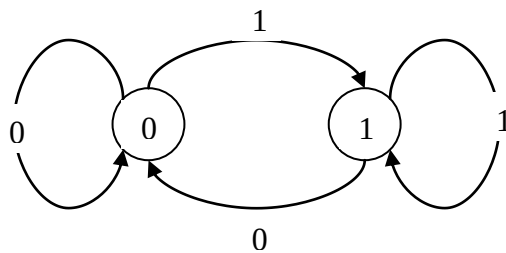
Доказательство:

Пусть $L_i(V)$ не пустое множество, для некоторого числа i и автомата V . Тогда существует $\beta \in L_i(V)$. Согласно предыдущей теореме, β «продолжаемо», т.е. найдется буква b выходного алфавита автомата V такая, что $\beta b \in L_i(V)$. Тогда и βb является продолжаемым. Повторяя конструкцию, получаем бесконечность $L_i(V)$. ■

Каждый конечный детерминированный инициальный автомат V задает, в общем случае, бесконечную цепочку вложенных регулярных, продолжаемых языков

$$L_1(V) \supseteq L_2(V) \supseteq L_3(V) \supseteq \dots$$

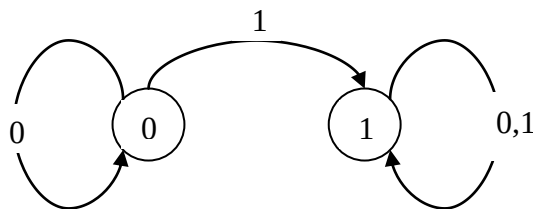
Для автомата «0-задержки», такая цепочка оказывается конечной:



$$\begin{aligned} L_1(V) &= \{ 0((0)^*(1)^*)^* \}; \\ L_2(V) &= L_1(V) = \{ 0((0)^*(1)^*)^* \}; \\ L_i(V) &= \emptyset, \text{ для } i > 2. \end{aligned}$$

Рисунок 2. Нулевая задержка и соответствующая ей цепочка регулярных языков.

Приведем пример автомата с бесконечной цепочкой:



$$\begin{aligned} L_1(V) &= \{ 0(0)^*(1)^* \}; \\ L_2(V) &= L_1(V) = \{ 0(0)^*(1)^* \}; \\ L_3(V) &= \{ 0(0)^*1(1)^* \}; \\ L_i(V) &= \{ 0(0)^*1^{\lfloor \log_2(i) \rfloor}(1)^* \}, \text{ для } i > 3. \end{aligned}$$

Рисунок 3. Простейший автомат с бесконечной цепочкой.

Под $(a)^*$ здесь понимается повторение символа a произвольное число раз, в том числе, пустое слово. Имеет место

Утверждение 1.2:

Если автомат $V = (A, Q, B, \varphi, \psi, q_0)$ таков, что для любого натурального i : множество $L_i(V)$ -не пусто, то для всякого r найдется $s > r$, такое что $L_s(V) \neq L_r(V)$.

В самом деле, пусть найдется p : $L_p(V) = L_{p+i}(V)$, для любого i натурального. Рассмотрим слово α : $|\alpha| = \min_{\beta \in L_p} |\beta|$.

$$\kappa_V(\alpha) < |A|^{|\alpha|} + 1.$$

Пусть теперь $i = |A|^{|\alpha|} + 1 - p$, тогда $\alpha \notin L_{p+i}(V)$. Противоречие.

Следующий пример показывает, что из равенства $L_i(V) = L_{i+1}(V)$ не следует, вообще говоря, $L_{i+1}(V) = L_{i+2}(V)$. В самом деле: зададим автомат V диаграммой Мура:

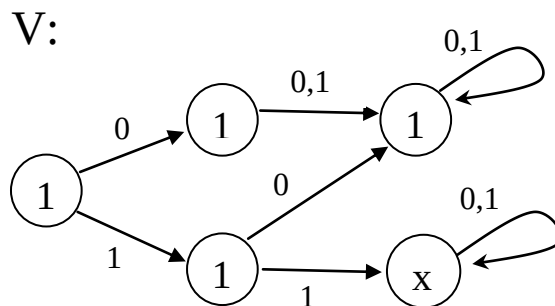


Рис 4. Автомат V.

Получаем:

$$L_1(V) = \{1, 11, 11(0^*1^*)^*\}, \quad L_2(V) = \{1, 11^*\}, \quad L_3(V) = \{111^*\}, \quad L_4(V) = \{111^*\},$$

$$L_5(V) = \{1111^*\} \neq L_4(V), \dots :$$

$$L_1(V) \supset L_2(V) \supset L_3(V) \equiv L_4(V) \supset L_5(V) \dots$$

Утверждение 1.3:

Для любого автомата $V = (A, Q, B, \varphi, \psi, q_0)$ выполнено $\bigcap_{i=1}^{\infty} L_i(V) = \emptyset$.

В самом деле: пусть верно обратное и найдется слово $\alpha \in \bigcap_{i=1}^{\infty} L_i(V)$. Тогда

$\kappa_V(\alpha) \geq N$, для любого наперед заданного натурального N . Что невозможно.

Следующий пример показывает, что из совпадения цепочек

$$L_1(V_1) \equiv L_1(V_2) \supseteq L_2(V_1) \equiv L_2(V_2) \supseteq L_3(V_1) \equiv L_3(V_2) \supseteq \dots$$

двух автоматов V_1, V_2 не следует их эквивалентность.

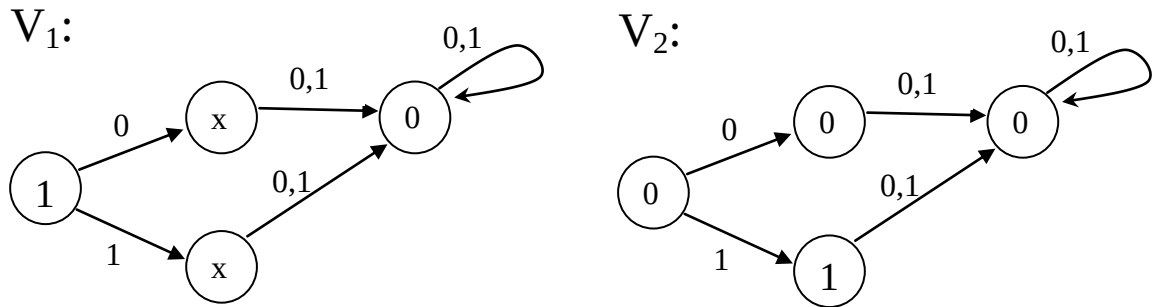


Рис 5. Диаграммы Мура автоматов V_1, V_2 .

Покажем, что гистограммные функции $\kappa_1 \equiv \kappa_2$. Действительно, $\kappa_1(1) = \kappa_2(1) = 2$, $\kappa_1(10) = \kappa_1(11) = \kappa_2(10) = \kappa_2(11) = 2$, и все слова длины более 2 ведут в эквивалентные состояния в автоматах V_1 и V_2 , что влечет совпадение значений гисторгаммных функций на словах длины 3 и более. Таким образом, $\kappa_1 \equiv \kappa_2$, однако, автоматы V_1 и V_2 имеют неэквивалентные состояния.

Зафиксируем натуральное n . Приведем пример двух цепочек языков, порождённых разными автоматами и совпадающих на длине 2^n . Рассмотрим автоматы V_3, V_4 с n состояниями, диаграммы Мура которых даны на рисунке 6.

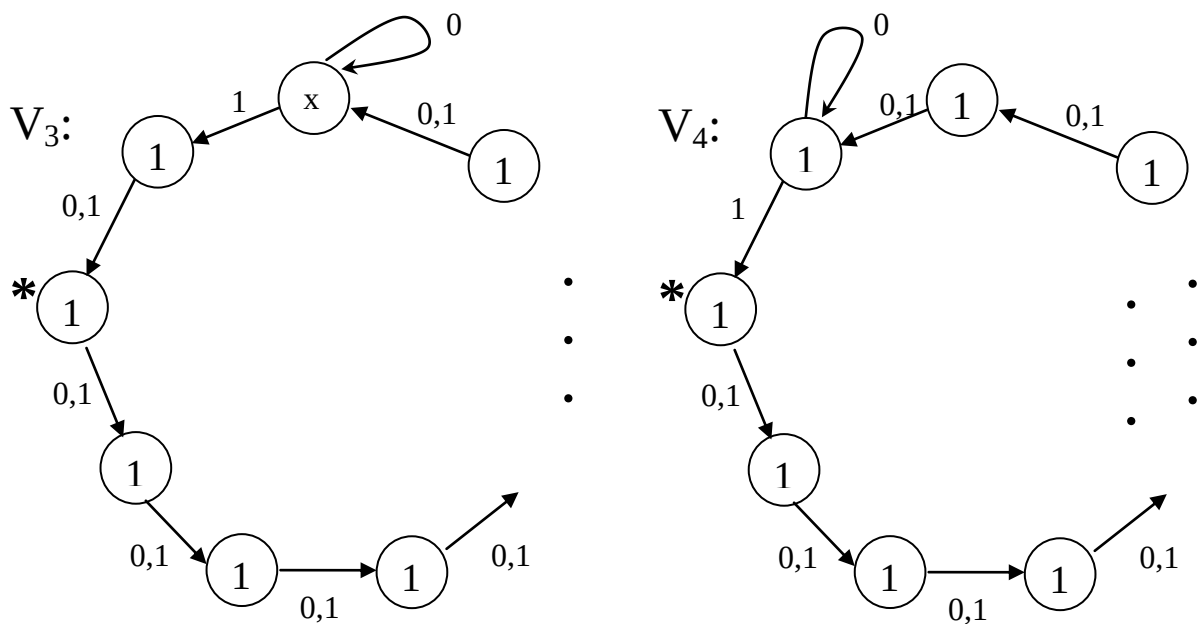


Рис 6. Диаграммы Мура автоматов V_3, V_4 .

Для автоматов V_3, V_4 истинно:

$$L_1(V_1) \equiv L_1(V_2) \supseteq L_2(V_1) \equiv L_2(V_2) \supseteq L_3(V_1) \equiv L_3(V_2) \supseteq \dots \supseteq L_{2^n}(V_1) \neq L_{2^n}(V_2) \dots$$

1.2 Классы языков, порожденные гистограммной функцией автомата.

Полагая фиксированными входной и выходной алфавиты A, B , $|A| \geq |B| > 1$, для фиксированного натурального i введем класс языков $\mathcal{L}_i$.

Определение 1.3:

$\mathcal{L}_i = \{ L_i(V) \mid V \in K(A, B) \}$, где $K(A, B)$ – класс всех автоматов с

входными и выходными алфавитами A, B , $|A| \geq |B| > 1$, соответственно.

Теорема 1.1 утверждает, что для любых натуральных i , любой язык $L \in \mathcal{L}_i$ является регулярным. Обратное, вообще говоря, неверно. Рассмотрим, к примеру, $A=B=E_2$ и регулярный язык

$$\{00, 01, 10, 11, 111^*, 101^*, 010^*, 000^*\}.$$

Из мощностных соображений, он не принадлежит классу языков $\mathcal{L}_i$ ни для какого $i > 1$. Действительно, принадлежность данного регулярного языка к классу $\mathcal{L}_i$ означает, что существует конечный детерминированный автомат $V = (E_2, Q, E_2, \varphi, \psi, q_0)$, такой что $\kappa_V(00) = \kappa_V(01) = \kappa_V(10) = \kappa_V(11) = i$. Слов длины 2 в алфавите E_2 ровно 4, когда как мощность образа $4i$ и совпадение возможно лишь при $i = 1$.

Известно, что класс регулярных языков замкнут относительно теоретико-множественных операций, конкатенации, итерации и взятия автоматного образа. Вопрос о замкнутости введенных классов языков относительно вышеперечисленных операций исследован в следующей теореме.

Теорема 1.3:

1. Класс языков $\mathcal{L}_i$ замкнут относительно операций: объединения, взятия образа автомата, конкатенации и итерации, но не замкнут относительно пересечения.
2. Класс языков $\mathcal{L}_i$ при $i > 1$ замкнут относительно операции взятия образа автомата, но не замкнут относительно теоретико-множественных операций объединения и пересечения.

Доказательство:

Пусть $B = \{b_1, \dots, b_p\}$. Для произвольного множества слов Φ через $\text{Pref}(\Phi)$ обозначим множество всех непустых начал слов из Φ . Из описания класса автоматно-перечислимых множеств в [4] следует, что для любых множеств слов $\Phi, \Theta \in \mathcal{L}_i$, выполнено $\text{Pref}(\Phi) \subseteq \Phi$, $\text{Pref}(\Theta) \subseteq \Theta$. Откуда

$$\text{Pref}(\Phi \cup \Theta) \subseteq \text{Pref}(\Phi) \cup \text{Pref}(\Theta) \subseteq \Phi \cup \Theta.$$

Более того, любое слово $\alpha \in \Phi \cup \Theta$ является началом некоторого другого слова $\alpha' \in \Phi$ или $\alpha' \in \Theta$, т.е оно будет началом слова $\alpha' \in \Phi \cup \Theta$. Учитывая

регулярность Φ, Θ имеем $\mathcal{L}$ замкнуто, относительно объединения. Так как суперпозиция автоматных функций является автоматной функцией, имеем $\Phi \in \mathcal{L}$, следовательно, $\Phi = f(A^*)$, $g(\Phi) = g(f(A^*)) = h(A^*)$. Поэтому для произвольной автоматной функции g : $g(\Phi) \in \mathcal{L}$.

Покажем, что для любых $\Phi, \Theta \in \mathcal{L}$, верно $\Phi\Theta \in \mathcal{L}$. Для этого, согласно описанию автоматно-перечислимых множеств надо установить:

- 1) $\Phi\Theta$ – префиксный язык.
- 2) $\Phi\Theta$ – регулярный язык.
- 3) $\Phi\Theta$ – продолжаемый язык

Свойство регулярности конкатенации языков, вытекает из того, что $\Phi, \Theta \in \mathcal{L} \subseteq \text{Reg}$. Префиксность конкатенации языков есть следствие префиксности Φ, Θ . Далее, если $\gamma \in \Phi\Theta$, то $\gamma = \alpha\beta$, где $\alpha \in \Phi$, $\beta \in \Theta$. В силу автоматной перечислимости Θ , β - начало некоторого $\beta' \in \Theta$, и γ суть начало $\alpha\beta' \in \Phi\Theta$. Если $\gamma = \alpha\beta$ и β - пустое слово, то в силу автоматной перечислимости Φ , найдется α' , что α - суть его начало. Т.е. в любом случае $\Phi\Theta$ будем продолжаемым языком.

Без изменения хода рассуждений, можно показать префиксность, регулярность и продолжаемость Φ^* , где $\Phi \in \mathcal{L}$. Аналогично получаем замкнутость $\mathcal{L}$ относительно операции итерации. Далее заметим, что любое множество $\Phi \in \mathcal{L}$ является бесконечным. Пусть $\Phi = \{b_1, b_1(b_1)^*\}$, $\Theta = \{b_1, b_1(b_2)^*\}$. Очевидно $\Phi, \Theta \in \mathcal{L}$. Но $\Phi \cap \Theta = \{b_1\} \notin \mathcal{L}$.

Пусть теперь $i > 1$. Рассмотрим такое $n \in N$, что

$$p^n > i \geq p^{n-1}, \text{ где } p = |B|$$

Покажем, что найдутся два автомата V, W , что

$$L_i(V) \cup L_i(W) \neq L_i(Z)$$

для любого автомата Z . Рассмотрим лексикографически упорядоченное множество слов

$$B^n = \{b_1^n, b_1^{n-1}b_2, \dots, b_p^n\} = \{\beta_1, \beta_2, \dots, \beta_p^n\},$$

где $B=\{b_1, \dots, b_p\}$. Пусть s_1 – суть максимальное число подряд идущих слов из B^n , которое может принадлежать $L_i(V)$, для автомата V . Ясно, что

$$s_1 < p^n,$$

т.к. у каждого слова из $\{\beta_1, \beta_2, \dots, \beta_{s_1}\}$ должен быть прообраз, и, следовательно $is_1 \leq p^n$. Аналогично, s_2 – максимальное число подряд идущих слов $\{\beta_{p^n-s_2+1}, \dots, \beta_{p^n}\}$, которое может принадлежать $L_i(W)$, для автомата W . Пусть автоматы V и W таковы, что $L_i(V), L_i(W)$ содержат

$$\{\beta_1, \beta_2, \dots, \beta_{s_1}\}$$

и

$$\{\beta_{p^n-s_2+1}, \beta_{p^n-s_2+2}, \dots, \beta_{p^n}\},$$

соответственно. Тогда $L_i(V) \cup L_i(W)$ содержит

$$\{\beta_1, \beta_2, \dots, \beta_{s_1}, \beta_{p^n-s_2+1}, \beta_{p^n-s_2+2}, \dots, \beta_{p^n}\}.$$

Из неравенства $s_1 < p^n$, вытекает, что

$$\rho_d(\beta_{s_1}, \beta_{p^n}) \geq \rho_d(\beta_{s_1}, \beta_{s_1+1}).$$

Следовательно, если предположить, что найдется автомат Z , что

$$\{\beta_1, \beta_2, \dots, \beta_{s_1}, \beta_{p^n-s_2+1}, \beta_{p^n-s_2+2}, \dots, \beta_{p^n}\} \subseteq L_i(Z),$$

то должен найтись автомат Z' , что

$$\{\beta_1, \beta_2, \dots, \beta_{s_1}, \beta_{s_1+1}\} \subseteq L_i(Z'),$$

согласно лемме о сжимающем отображении (см. главу 2). Тем самым, s_1 не является максимальным. Полученное противоречие доказывает незамкнутость L_i относительно объединения.

Покажем теперь, что найдутся два автомата V, W , что

$$L_i(V) \cap L_i(W) \neq L_i(Z)$$

для любого автомата Z . Пусть теперь

$$L_i(V) = \{\beta, \beta\delta_1, \beta\delta_2, \dots\},$$

$$L_i(W) = \{\beta, \beta\gamma_1, \beta\gamma_2, \dots\},$$

где для любого натурального j , слово δ_j не содержит b_1 , а γ_j содержит хотя бы одну букву b_1 , где $b_1 \in B = \{b_1, \dots, b_p\}$. Легко видеть, что такие автоматы существуют. По построению $L_i(V) \cap L_i(W) = \{\beta\}$. Однако, из *Теоремы 1.2*, следует, что для любого автомата D : $L_i(D)$ является бесконечным множеством, либо пустым. Тем самым $\mathcal{L}$ не замкнут относительно операции $\cap$.

Замкнутость относительно автоматаного образа вытекает из детерминированности словарного автоматного отображения ■.

Самостоятельный интерес представляет связь языков класса $\mathcal{L}$ с другими типами регулярных языков. Класс *Pref* всех префиксных языков содержит подкласс *Fact* так называемых *факториальных* языков, т.е. языков, замкнутых относительно операции взятия подслов. Известно [5], что каждый факториальный язык L однозначно задается своим анτισловарем M . Более того, если анτισловарь M языка L алфавита B конечен, то из равенства

$$L = B^* \setminus B^* M B^*$$

следует регулярность языка L . Такие языки называются *языками с конечным анτισловарем* и обозначаются *FAD*. Язык L совпадающий с некоторым анτισловарем факториального языка называется *антифакториальным*. Класс всех антифакториальных языков обозначим через *AFact*. Наконец через *Reg* обозначим класс всех регулярных языков. Согласно [7] обозначим

$Pref = \{ L \mid \text{если } \alpha\alpha \in L, \text{ то } \alpha \in L, \text{ для любого слова } \alpha \}.$

$Fact = \{ L \mid \text{если } \alpha\beta\gamma \in L, \text{ то } \beta \in L, \text{ для любых (м.б. пустых) слов } \alpha, \beta, \gamma \}.$

$FAD = \{ L \mid M = \{ \alpha \notin L \mid \text{никакое подслово слова } \alpha \text{ не обладает этим свой-ом} \} \mid |M| < \infty \}.$

$AFact = \{ L \mid \text{если } \beta \in L, \text{ то для любых } (\alpha\gamma \neq \Lambda) \text{ слов } \alpha, \gamma: \alpha\beta\gamma \notin L \}.$

Следующую Теорему 1.4 удобно представлять в виде схемы (рис.7):

Теорема 1.4:

Для любых алфавитов A, B , $|A| \geq |B| > 1$, выполнены следующие утверждения:

- 1) $\mathcal{L} \cup (\text{Reg} \cap \text{ACont} \cap \text{Pref}) = \text{Reg} \cap \text{Pref}$.
- 2) Для любого натурального n : $\bigcap_{i=1}^n \mathcal{L}_i \neq \emptyset$, но $\bigcap_{i=1}^{\infty} \mathcal{L}_i = \emptyset$.
- 3) Для любого натурального i : $\mathcal{L}_i \cap \text{AFact} = \emptyset$.
- 4) Для любого натурального i : $\mathcal{L}_i \cap \text{FAD} \neq \emptyset$, $\mathcal{L}_i \cap \text{Fact} \neq \emptyset$, $\mathcal{L}_i \cap \text{Pref} \neq \emptyset$.

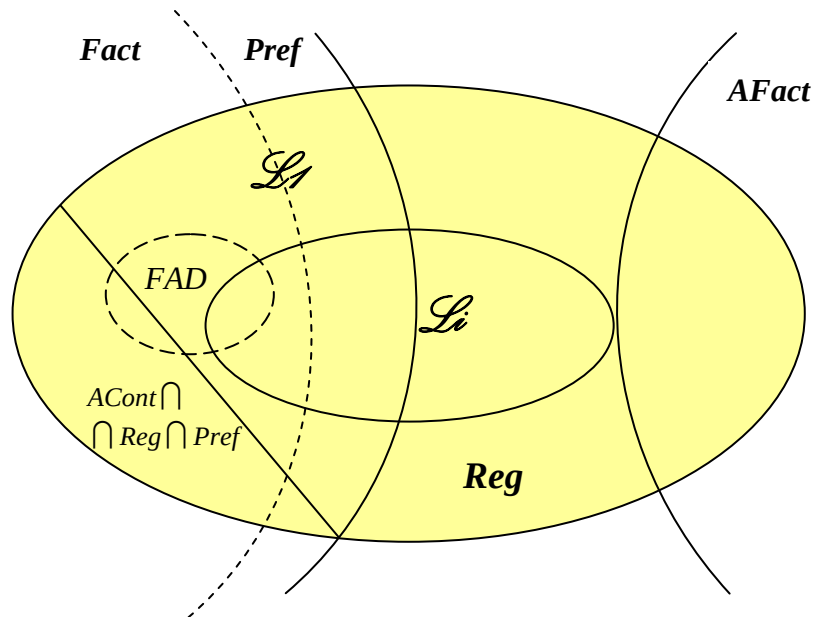


Рис 7. Взаиморасположение классов языков.

Доказательство:

Известно [1], что префиксный регулярный язык является автоматно-перечислимым в том и только том случае, если он продолжаем. Следовательно,

$$\mathcal{L} = (Reg \cap Cont \cap Pref),$$

учитывая, что $(ACont \cup Cont)$ суть все множество языков, получаем

$$\mathcal{L} \cup (Reg \cap ACont \cap Pref) = (Reg \cap Cont \cap Pref) \cup (Reg \cap ACont \cap Pref) = (Reg \cap Pref \cap (ACont \cup Cont)) = Reg \cap Pref.$$

Второе утверждение теоремы докажем от противного. Пусть $\bigcap_{i=1}^{\infty} \mathcal{L}_i$

$\neq \emptyset$ и слово β таково, что $\beta \in \bigcap_{i=1}^{\infty} \mathcal{L}_i$. Т.е. для любого натурального M ,

найдется автомат V : $\kappa_V(\beta) \geq M$. Обозначим длину слова β через m . Тогда, из мощностных соображений: $\kappa_V(\beta) \leq |A|^m$. Что противоречит

произвольности M . Далее заметим, что если для некоторого натурального M найдется автомат V и слово $\beta \in L_M(V)$, то

$$\beta \in L_M(V) \cap L_{M-1}(V) \cap \dots \cap L_1(V),$$

так как

$$L_M(V) \subseteq L_{M-1}(V) \subseteq \dots \subseteq L_1(V),$$

откуда $\{\beta\} \subseteq \bigcap_{i=1}^M \mathcal{L}_i$. Легко видеть, что для любого натурального M

существует автомат, например константный, и слово β для которых это будет выполнено.

Третье утверждение напрямую вытекает из *Теоремы 1.2*, т.к. любой антифакториальный язык по определению непродолжаем.

$$AFact \subseteq ACont,$$

но для любого натурального M : $\mathcal{L}_M \subseteq Cont$.

Для доказательства последнего утверждения, зафиксируем наперед произвольное натуральное M . Обозначим автомат-задержку (нулевую) через Z_0 , а автомат с одним состоянием и сюръективной выходной функцией $\psi: \{q_0\} \times A \rightarrow B$, через X_1 . Очевидно, такой автомат существует, т.к. $|A| \geq |B|$. Построим новый структурный автомат, последовательно соединяя $\log_{|B|}(M) + 1$ задержек с автоматом X_1 , как показано на рисунке.

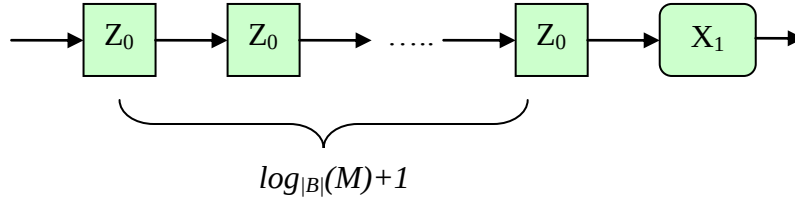


Рис 8. Построение автомата V , такого, что $L_M(V) \in FAD$.

Соответствующий построенному структурному автомату, автомат V обладает, очевидно, свойством $L_M(V) \in FAD$, так как антисловарь языка $L_M(V)$ состоит из слов, длины меньшей $M+1$, и, следовательно, ограничен. Так как класс $FAD \subseteq Fact \subseteq Pref$, то из произвольности M следует $\mathcal{L}_M \cap FAD \neq \emptyset$, $\mathcal{L}_M \cap Fact \neq \emptyset$, $\mathcal{L}_M \cap Pref \neq \emptyset$. ■.

1.3 Критерий принадлежности языка к классу $\mathcal{L}_p$.

Выше было установлено, что каждый язык типа $\mathcal{L}_p$ регулярен, но оставался вопрос проверяемо ли свойство произвольного регулярного языка быть языком L_p -типа. Эта проблема решена в данном параграфе.

Ранее было доказано, что не каждое регулярное продолжаемое множество является языком типа $\mathcal{L}_p$.

Для натурального p введем понятие p -раскраски автомата без выхода, представляющего событие с помощью финальных состояний.

Определение 1.4:

Пусть задан автомат

$$W = (B, Q, \varphi, q_0, Q_F) \quad (2)$$

без выхода, с выделенным множеством финальных состояний Q_F .

Введем функцию

$$k: Q \rightarrow \mathbb{N} \cup \{0\},$$

сопоставляющую каждому состоянию из Q неотрицательное число. Будем говорить, что функция k задает на автомате V правильную p -раскраску состояний, если для любой вершины $q \in Q$ выполнены:

$$1) \quad k(q) < p, \quad \text{для любого } q \notin Q_F,$$

$$k(q) \geq p, \quad \text{для любого } q \in Q_F \text{ и}$$

$$k(q_0) = 1.$$

$$2) \quad k(q) \cdot |B| = \sum_{i=1}^{|B|} k(\varphi(q, b_i)), \quad \text{если } \forall i \quad \varphi(q, b_i) \notin Q_F \text{ и}$$

$$k(q) \cdot |B| \geq \sum_{i, \varphi(q, b_i) \notin Q_F} k(\varphi(q, b_i)) + s(q) \cdot p, \quad \text{где } s(q) = |B| - \left| \{i \mid \varphi(q, b_i) \notin Q_F\} \right|$$

Автомат V , в таком случае назовем правильно p -раскрашенным.

Заметим, что не всякий автомат типа (2) может быть правильно p -раскрашен. Далее, если не оговорено иное, под словами язык L представлен автоматом V , подразумевается, что автомат без выхода представляет язык с помощью множества финальных состояний. А именно, $L = \{\alpha \mid \varphi(q_0, \alpha) \in Q_F\}$.

Теорема 1.5:

Пусть $p > 1$, автомат $V = (B, Q, \varphi, q_0, Q_F)$ представляет язык L . Тогда $L \in \mathcal{L}_p$ в том и только том случае, когда найдется правильно p -раскрашенный автомат

$$V' = (B, Q', \varphi', q_0', Q_F'),$$

также, представляющий L . Причем число состояний нового автомата сравнимо с числом состояний исходного, а именно:

$$|Q'| \leq \frac{p(p-1)}{2} \cdot |Q \setminus Q_F| + p \cdot |Q_F|.$$

Следствие 1.2:

Язык L , представимый сильносвязным автоматом, не лежит в $\mathcal{L}_p$, ни для какого $p > 1$.

Используя мощностные соображения, можно доказать

Утверждение 1.4:

Если $V = (B, Q, \varphi, q_0, Q_F)$ представляет некоторый язык $L \in \mathcal{L}_p$, то

$$p \leq |B|^{|Q|-1}.$$

Обозначим через $Reg = Reg(B)$ все регулярные языки над алфавитом B и пусть

$$\mathcal{L}_0 = Reg \setminus \bigcup_{i=1}^{\infty} \mathcal{L}_i \text{ ываыва}$$

В [21] показано, что для произвольных натуральных $p \neq q$, $\mathcal{L}_p \not\subseteq \mathcal{L}_q$. Таким образом, система языков

$$\{\mathcal{L}_0, \mathcal{L}_1, \mathcal{L}_2, \dots\}$$

образует неизбыточное покрытие множества регулярных языков алфавита B . Имеет место

Теорема 1.6:

Существует алгоритм, который для любого языка $L \in Reg$ определяет все те p , что $L \in \mathcal{L}_p$.

Доказательства Теорем 1.5, 1.6 даны в следующем параграфе.

1.4 Доказательство критерия принадлежности языка к классу $\mathcal{L}_p$. (теоремы 1.5, 1.6)

Формализуем определение источника на языке множеств и отношений.

Под источником понимается пятерка $V' = (B, Q, \lambda, q_0, Q_F)$, где B – выходной алфавит, Q – конечное множество состояний, отношение λ задано на декартовом произведении $\lambda \subseteq Q \times Q \times B \times \mathbb{N}_0$, q_0 – начальное состояние, Q_F – множество финальных. Если $(q_1, q_2, b, s) \in \lambda$, то в источнике имеется s

кратных ребер (q_1, q_2) , которым приписан символ b . Таким образом, каждому пути в источнике единственным образом соответствует слово из B^* , однако, одному слову может соответствовать несколько различных путей. Множество слов, ведущих из q_0 в некоторое $q \in Q_F$, образуют язык, представляемый источником V' . Здесь и далее будем требовать, чтобы из каждого состояния источника выходило ровно $|B|$ ребер.

Пусть задан лес F у которого каждому ребру приписана некоторая буква b алфавита B .

Множеством слов перечислимым источником V' назовем $L_p(V') = \{ \beta \in B^* \mid \text{в } V' \text{ найдется не менее } p \text{ различных путей из начального состояния } q_0 \text{ в одно из финальных, которым приписано } \beta \}$.

Множеством слов перечислимым лесом F назовем $L_p(F) = \{ \beta \in B^* \mid \text{в } F \text{ найдется не менее } p \text{ различных путей, которым приписано } \beta, \text{ начинающихся в корнях деревьев леса.} \}$.

Назовем леса F_1 и F_2 p -эквивалентными, если $L_p(F_1) = L_p(F_2)$. Эквивалентные, леса порождают одинаковое множество слов кратности большей либо равной p .

Через D обозначим информационное дерево автомата V , ребрам которого приписаны только выходные символы алфавита B . Продемонстрируем идею доказательства теоремы 1 на простом примере. Пусть $V = (E_2, \{q_0, q_1, q_2\}, \varphi, q_0, Q_F)$ правильно 2-раскрашенный автомат над алфавитом $E_2 = \{0, 1\}$, и раскраска задается функцией

$$k: k(q_0)=1, k(q_1)=0, k(q_2)=2.$$

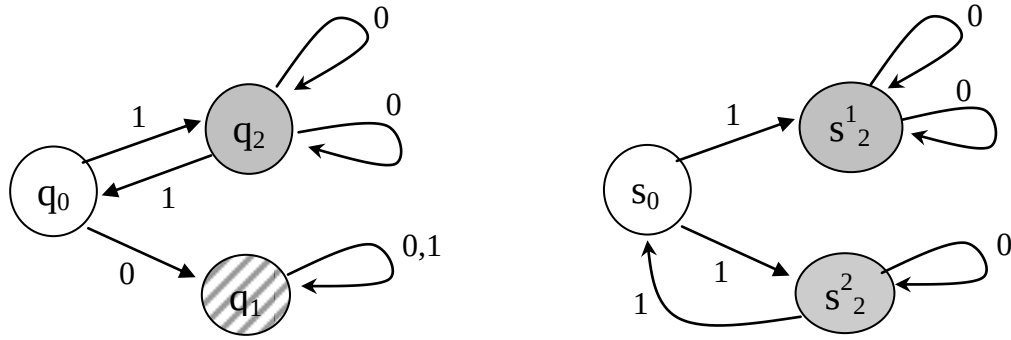


Рисунок 9. Автомат V с раскрашенными состояниями.

Пусть V задает регулярный язык L с помощью состояния q_2 множества состояний. $L = \{1(0^*(11)^*)^*\}$. Покажем, что $L \in \mathcal{L}_2$, для чего предъявим источник W : $L = L_2(W)$.

Используя диаграмму переходов автомата V , построим источник W : пусть каждому состоянию $q \in \{q_0, q_1, q_2\}$ соответствует набор $Q(q) = \{s^1_q, \dots, s^M_q\}$ состояний источника W , причем $M = k(q)$. Таким образом, в новом источнике будет 1 состояние, «порожденное» q_0 , и 2 состояния, «порожденные» q_2 , выделенные на рисунке 9 слева соответствующим цветом. Так как теперь буква $1 \in E_2$ такова, что $\varphi(q_0, 1) = q_2$, то соединим наборы состояний $Q(q_0) = \{s_0\}$ и $Q(q_2) = \{s^1_2, s^2_2\}$, проведя $k(q_2) = 2$ нагруженных символом 1 ребра из s_0 в s^1_2, s^2_2 . Аналогично для q_2 и символа 0 и 1. Полученный источник W удовлетворяет: $L_2(W) = \{1(0^*(11)^*)^*\} = L$.

Покажем теперь, что если язык $L = L_4(W)$ для источника W , и автомат $V = (E_3, \{q_0, q_1, q_2, q_3, q_4\}, \varphi, q_0, Q_F)$ над $E_3 = \{0, 1, 2\}$ представляет L с помощью множества финальных состояний $Q_F = \{q_3\}$, то найдется автомат V' , представляющего язык L с помощью финальных состояний, который можно правильно p -раскрасить. Диаграмма автомата V представлена на рисунке:

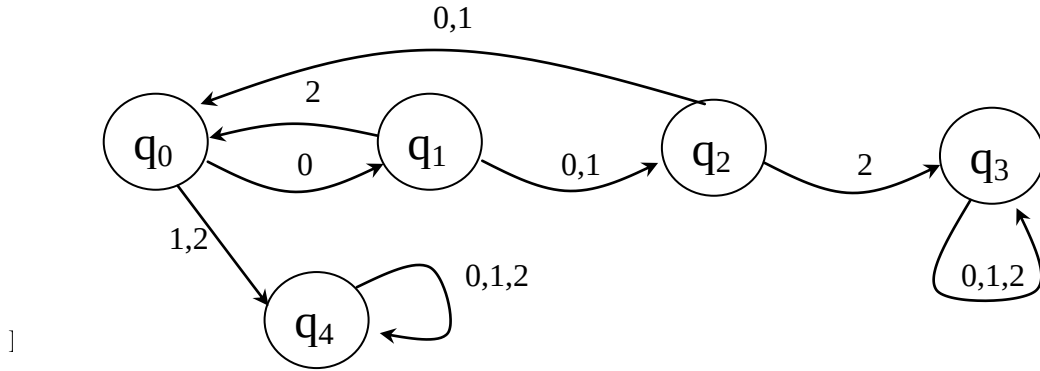


Рисунок 10. Автомат V допускает правильную p -раскраску.

Далее, по информационному дереву D источника W (каждому ребру приписана буква алфавита E_3) и автомату V будет построен искомый автомат V' . Каждому пути в информационном дереве D , соответствует слово $\beta \in E_3^*$, которое однозначно определяет состояние q автомата V , что $\varphi(q_0, \beta) = q$. Используя это правило, припишем каждой вершине дерева D соответствующее состояние из $\{q_0, q_1, q_2, q_3, q_4\}$. Следовательно, каждое слово $\beta \in E_3^*$ задает не только состояние $q = \varphi(q_0, \beta)$, но и число $m = m(q, \beta)$ путей по слову β из корня дерева D в вершины, которым приписано состояние q . Рассмотрим все возможные пары (q, m) , отождествим пару (q, m_1) с (q, m_2) , если $m_1 > m_2 \geq p$; Таким образом, возможных сочетаний (q, m) конечное число. Пусть теперь множество состояний автомата V' образуют ровно те сочетания (q, m) , которые встречаются в нагруженном дереве D . В рассматриваемом примере множество состояний автомата V' это:

$$S_0 = (q_0, 1), S_1 = (q_1, 2), S_2 = (q_2, 2), S_3 = (q_2, 3), S_4 = \{(q_3, m) \mid m \geq 4\}, S_5 = (q_4, 1).$$

Именно натуральное $m = m(q, \beta)$ в паре (q, m) задает цвет состояния нового автомата. На рисунке 11, состояние S выделено тем же цветом, что и порождающая его пара на дереве. Заметим, что дерево D естественным образом задает способ соединения полученных состояний. Таким

образом строится автомат $V' = (E_3, \{S_0, S_1, S_2, S_3, S_4, S_5\}, \varphi, S_0, S_4)$ представляющий язык L с помощью финального состояний S_4 . Осталось заметить, что V' можно правильно 4-раскрасить. Пусть k – функция цвета состояния, тогда:

$$k(S_0)=1, k(S_1)=2, k(S_2)=2, k(S_3)=3, k(S_4)=4, k(S_5)=0.$$

Немаловажно заметить, что исходный автомат V обладал меньшим числом состояний, но не допускал правильной 4-раскраски.

Перейдем к доказательству **Теоремы 1.5**:

Достаточность.

Пусть автомат $V = (B, Q, \varphi, q_0, Q_F)$ имеет правильную раскраску $k: Q \rightarrow Z_+$. Покажем, что существует автомат $V_1 = (B, Q_1, B, \varphi_1, \psi_1, q_0'')$, что $L = L_p(V_1)$. Очевидно для этого достаточно доказать существование источника W с тем же свойством $L = L_p(W)$, у которого бы из каждого состояния выходило ровно $|B|$ ребер. Сопоставим каждому состоянию $q_i \in Q \setminus Q_F$ набор $\tilde{Q}(q_i) = \{q_i', \dots, q_i^{k(q_i)}\}$, а если $q_i \in Q_F$, то $\tilde{Q}(q_i) = \{q_i', \dots, q_i^p\}$. Элементы из введенных наборов будут состояниями источника W .

Пусть, без ограничения общности, q_i таково, что:

$$\begin{aligned} \varphi(q_i, b_1) &= \varphi(q_i, b_2) = \dots = \varphi(q_i, b_{s(1)}) = q_{t(1)}, \\ \varphi(q_i, b_{s(1)+1}) &= \varphi(q_i, b_{s(1)+2}) = \dots = \varphi(q_i, b_{s(2)}) = q_{t(2)}, \\ &\dots \\ \varphi(q_i, b_{s(m-1)+1}) &= \dots = \varphi(q_i, b_{s(m)}) = q_{t(m)}. \end{aligned}$$

где для каждого j $1 \leq s(j) \leq n$ и $\sum_{i=1}^m s(i) = |B|$. Соединим набор состояний

$\tilde{Q}(q_i)$ $|B|k(q)$ ребрами с наборами $\tilde{Q}(q_{t(1)}), \dots, \tilde{Q}(q_{t(m)})$, так, чтобы в каждое состояние набора $\tilde{Q}(q_{t(j)})$ для любого j входило 1 ребро, если $q_{t(j)} \in Q \setminus Q_F$. В случае $q_{t(j)} \in Q_F$ потребуем, чтобы в первое состояние $q_{t(j)}'$ набора $\tilde{Q}(q_{t(1)})$ входило не менее 1 ребра. Также, если $q_i \in Q_F$, то первое

состояние q_i' набора $\tilde{Q}(q_i)$ соединим с только с первыми состояниями наборов, соответствующих финальным состояниям. Хотя бы 1 такое найдется, так как из условия существования правильной раскраски следует, что, если $q_i \in Q_F$, то существует $r: q_{i(r)} \in Q_F, r \leq m$. В противном случае по свойству 2 правильной раскраски получалось бы $k(q)|B| < p|B|$, а по свойству 1 - противоречащее $k(q)|B| = p|B|$. Также будем требовать, чтобы из каждого состояния набора $\tilde{Q}(q_i)$ выходило ровно $|B|$ ребер. Несложно видеть, что в случае существования правильной раскраски все эти требования выполнимы.

Итак, если состояния q_1, q_2 автомата V таковы, что $\varphi(q_1, a) = q_2$, то в источнике W им соответствуют наборы $\tilde{Q}(q_1)$ и $\tilde{Q}(q_2)$, состоящие из $k(q_1)$ и $k(q_2)$ копий состояний q_1, q_2 , соответственно. В каждое состояние набора $\tilde{Q}(q_2)$ за исключением, быть может первого, входит единственное ребро, выходящее из состояний набора $\tilde{Q}(q_1)$, которому приписана буква a . В первое же состояние набора $\tilde{Q}(q_2)$ входит одно ребро, если $q_2 \notin Q_F$ и не менее одного, в случае $q_2 \in Q_F$. Значит возможны 4 случая:

А) $q_1 \notin Q_F, q_2 \notin Q_F$. Пусть некоторое слово $\beta: \varphi(q_0, \beta) = q_1$ ведет в каждое состояние набора $\tilde{Q}(q_1)$. В W не существует иных состояний, кроме принадлежащих этому набору, в которые «вело» бы слово β , так как таковых не существует и в автомате V . Принимая во внимание тот факт, что слово βa однозначно определяет состояние q_2 , а, следовательно, и набор $\tilde{Q}(q_2)$, получаем, что найдется ровно $|\tilde{Q}(q_2)| = k(q_2) < p$ путей в W , которым приписано слово βa .

В) $q_1 \in Q_F, q_2 \notin Q_F$. Пусть некоторое слово $\beta: \varphi(q_0, \beta) = q_1$ таково, что в W существуют различные пути, которые соответствуют этому слову, ведущие в каждое состояние набора $\tilde{Q}(q_1)$. Причем, в первое состояние q_1' быть может, ведет сразу несколько таких путей. В W не

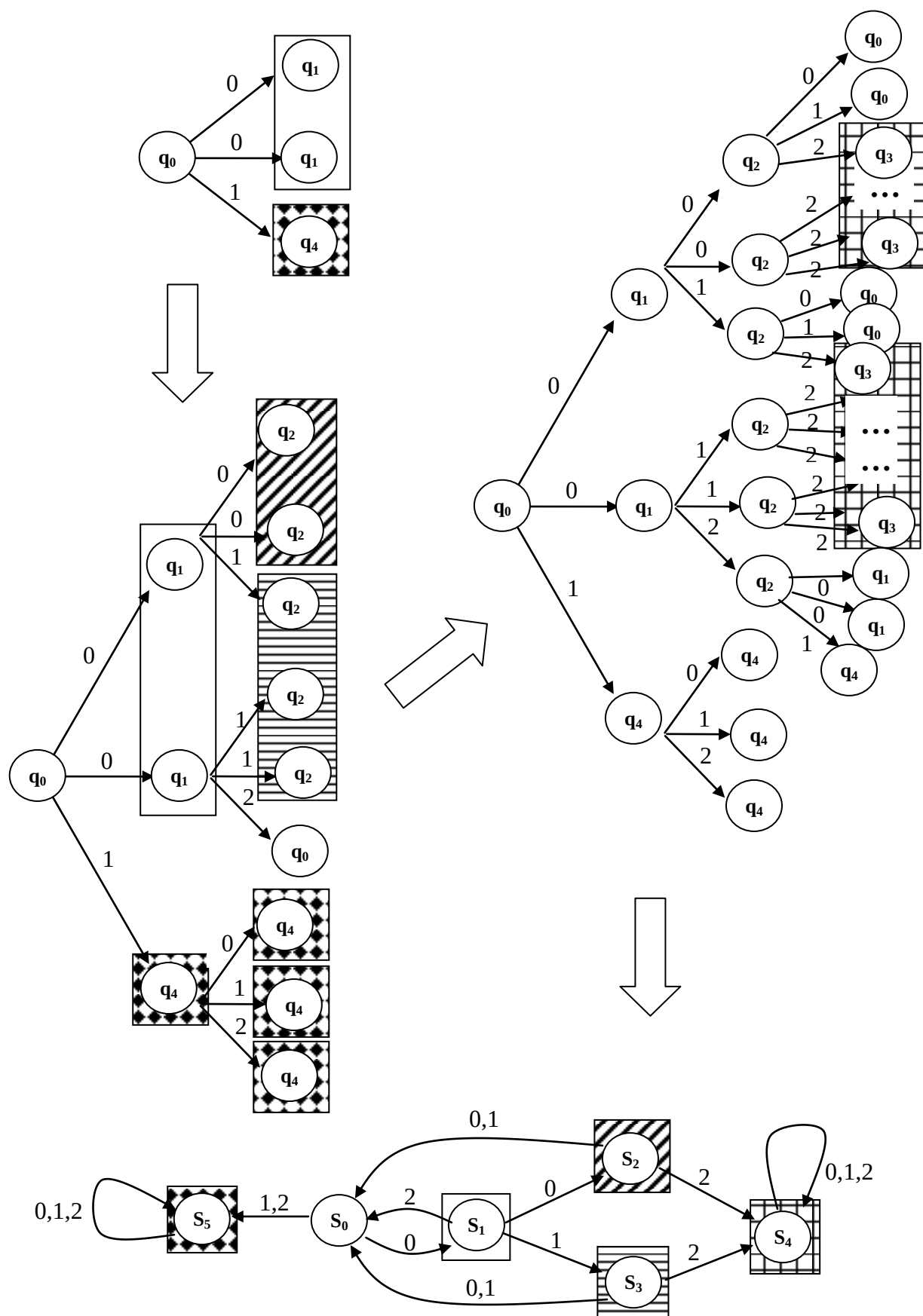


Рисунок 11. Построение правильно p -раскаршенного автомата V' по дереву D и автомату V . Начальное состояние - S_0 , финальное - S_4 .

существует иных состояний, кроме принадлежащих этому набору, в которые «вело» бы слово β , так как таковых не существует и в автомате V . По построению, из первого состояния q_1 набора $\tilde{Q}(q_1)$ не выходит ребер, ведущих в какое-либо состояние набора $\tilde{Q}(q_2)$. Значит в каждое состояние набора $\tilde{Q}(q_1)$, из которого выходят ребра в набор $\tilde{Q}(q_2)$, входит ровно 1 путь, соответствующий слову β . Следовательно в W ровно $|\tilde{Q}(q_2)| = k(q_2) < p$ путей, которым приписано слово βa .

C) $q_1 \notin Q_F, q_2 \in Q_F$. Совершенно аналогично, предполагая, что β : $\varphi(q_0, \beta) = q_1$ такое, что в W существуют различные пути, которые соответствуют этому слову, ведут в каждое состояние набора $\tilde{Q}(q_1)$ и таких путей ровно $k(q_1)$. Следовательно, путей, соответствующих слову βa не менее, чем $|\tilde{Q}(q_2)| = p$.

D) $q_1 \in Q_F, q_2 \in Q_F$. Абсолютно аналогично случаю B), получаем что количество путей в W , соответствующих слову βa , не менее чем $|\tilde{Q}(q_2)| = p$.

Осталось заметить, что для любого q , что найдется $b \in B$: $\varphi(q_0, b) = q$, верно: в каждое состояние набора $\tilde{Q}(q)$, кроме, быть может первого, входит ровно одно ребро с пометкой b из $\tilde{Q}(q_0)$, а в первое состояние может входить несколько. Это позволяет говорить, что $L_p(W) = L$. Замечу, что число состояний источника W , а следовательно и автомата V не превосходит $p|Q|$.

Необходимость.

Пусть теперь автомат $V = (B, Q, \varphi, q_0)$ задает с помощью набора финальных состояний Q_F язык $L \in \mathcal{L}_p$. Очевидно, достаточно рассматривать только автоматы без недостижимых состояний. Покажем,

что найдется автомат V'' , представляющий язык L и имеющий правильную p -раскраску $k: Q \rightarrow Z_+$. Из условия $L \in \mathcal{L}_p$ следует существование источника $V': L = L_p(V')$. Дерево этого источника обозначим через D' . Каждому ребру дерева естественным образом приписано некоторое слово из B . Так как произвольное слово $\beta \in B$ однозначно определяет состояние $q = \varphi(q_0, \beta)$ автомата V , каждой вершине дерева D' , в которую ведет путь из вершины, соответствующий слову β , можно однозначно сопоставить состояние автомата q . Корню дерева D' припишем q_0 . Нагруженное дерево D' задает функцию $\hat{k}: B^* \cdot Q \rightarrow \aleph$, где $\hat{k}(\beta, q)$ - суть число вершин дерева, с приписанным состоянием q в которые ведут пути из корня, соответствующие слову β . Другими словами, $\hat{k}(\beta, q)$ есть число корней деревьев, входящих в лес $F(\beta, q)$, где $F(\beta, q)$ – набор деревьев с корнями в вершинах D' , которым приписано состояние q и в которые ведут пути по слову β из корня D' . Если лес $F(\beta, q)$ пуст, считаем функцию $\hat{k}$ неопределенной на наборе (β, q) . Легко видеть, что для любого допустимого¹ $\beta \in B$: $0 < \hat{k}(\beta, q) < p$, если $q \notin Q_F$ и $\hat{k}(\beta, q) \geq p$, если $q \in Q_F$, так как условие $q \in Q_F$ эквивалентно существованию в дереве D' не менее p различных путей, которым приписано β . Аналогично, для произвольных (α, q') сопоставим лесу $F(\alpha, q')$ частично определенную функцию $k_{F(\alpha, q')}: B^* \rightarrow \aleph$, где $k_{F(\alpha, q')}(\gamma)$ есть количество путей исходящих из корней деревьев леса $F(\alpha, q')$, которым приписано слово $\gamma \in B^*$. Наконец заметим, что источник V' задает естественным образом частично определенную функцию $k: B^* \rightarrow \aleph$, где $k(\delta)$ есть количество путей в V' ведущих из его начального состояния, которым приписано слово $\delta \in B^*$.

Заметим теперь, что равенство $\hat{k}(\alpha, q) = \hat{k}(\beta, q)$ для некоторого $q = \varphi(q_0, \alpha) = \varphi(q_0, \beta)$ влечет за собой L_p -эквивалентность лесов $F(\alpha, q)$ и $F(\beta, q)$.

¹ Под допустимым словом понимается такое β , что $\hat{k}(\beta, q)$ определено. Ясно, что хотя бы одно такое слово всегда найдется, иначе автомат V имел бы недостижимое состояние q .

Действительно, для произвольного $\gamma \in B^*$: $k(\alpha\gamma) = k_{F(\alpha,q)}(\gamma)$. Равенство верно, так как все пути, выходящие из корня D' , инцидентные слову α ведут в лес $F(\alpha,q)$. Последнее истинно вследствие единственности такого $q \in Q$, что $q = \varphi(q_0, \alpha)$. Пусть теперь $k(\alpha\gamma) \geq p$, $\Rightarrow k_{F(\alpha,q)}(\gamma) \geq p$. Аналогично получаем

$k(\beta\gamma) \geq p$, $\Rightarrow k_{F(\beta,q)}(\gamma) \geq p$. Отметим, теперь, что условия $k(\alpha\gamma) \geq p$ и $k(\beta\gamma) \geq p$ эквивалентны. Фактически, если они выполняются/не выполняются одновременно, то найдутся $\alpha, \beta, \gamma \in B^*$: $\varphi(q_0, \alpha) = \varphi(q_0, \beta) = q$, и, следовательно, $\varphi(q_0, \alpha\gamma) = \varphi(q_0, \beta\gamma) = \varphi(q, \gamma)$, что означает, что дерево D' и источник V' не обладают свойством $L = L_p(V')$, т.к. $\alpha\gamma$ и $\beta\gamma$ или одновременно принадлежат Q_F или одновременно не принадлежат Q_F . Из эквивалентности условий $k(\alpha\gamma) \geq p$ и $k(\beta\gamma) \geq p$ получаем эквивалентность $k_{F(\alpha,q)}(\gamma) \geq p$ и $k_{F(\beta,q)}(\gamma) \geq p$, что означает L_p -эквивалентность лесов $F(\alpha,q)$ и $F(\beta,q)$.

Аналогично покажем, что из $\hat{k}(\alpha,q) \geq \hat{k}(\beta,q) \geq p$ следует L_p -эквивалентность лесов $F(\alpha,q)$ и $F(\beta,q)$. Предположим, что найдется слово γ : $k_{F(\alpha,q)}(\gamma) \geq p$, но $k_{F(\beta,q)}(\gamma) < p$. Так как слова $\alpha, \beta \in L_p$, то $\varphi(q_0, \alpha) = \varphi(q_0, \beta) = q \in Q_F$ по свойству автомата V . И в силу $k_{F(\alpha,q)}(\gamma) = k(\alpha\gamma) \geq p$, $\varphi(q, \alpha) \in Q_F$. Но, с другой стороны, $k_{F(\beta,q)}(\gamma) = k(\beta\gamma) < p$, и $\varphi(q, \gamma) \notin Q_F$, что невозможно. Из полученного противоречия вытекает эквивалентность условий $k_{F(\alpha,q)}(\gamma) \geq p$ и $k_{F(\beta,q)}(\gamma) \geq p$ для произвольного γ .

Начиная с первого будет последовательно рассматривать уровни дерева D' . Пусть сейчас наблюдается i -ый, где i – натуральное, и все различные слова $\{\beta_1, \dots, \beta_s\}$ алфавита B , ведущие в вершины этого уровня, $s \leq |B|^i$. Если β_j , $1 \leq j \leq s$ таково что существует β , $|\beta| < i$: $\varphi(q_0, \beta) = \varphi(q_0, \beta_j) = q$ и

$\hat{k}(\beta, q) = \hat{k}(\beta_j, q)$ или $\hat{k}(\beta, q) \geq \hat{k}(\beta_j, q) \geq p$ (т.е. $q \in Q_F$), то выкинем из рассмотрения все продолжения слова β_j , удалив из D' L_p -эквивалентный $F(\beta, q)$ лес $F(\beta_j, q)$. Число L_p -эквивалентных лесов вида $F(*, q)$ конечно, для

каждого $q \in Q$. Состояний автомата V также конечное число. Следовательно, найдется N -натуральное, что на N -ом уровне дерева D' уже не останется лесов, L_p -эквивалентные к которым не встречались бы на более ранних уровнях. Т.е. получим конечное дерево D_{cut} , которое естественным образом задает конечный источник V_{cut} со свойством $L_p(V_{cut}) = L_p(D')$, так как удалялись только «взаимозаменяемые» L_p -эквивалентные леса².

Каждому состоянию q из $Q \setminus Q_F$ в дереве D_{cut} соответствует набор различных чисел $\hat{k}(\beta_1, q), \dots, \hat{k}(\beta_m, q)$, где $\beta_1, \dots, \beta_m$ - некоторые слова алфавита B и $m \leq p(p-1)/2$. А каждому q' из Q_F - только одно число p . То есть дерево D_{cut} однозначно задает набор пар из $Q \times E_p: (q_0, 1), \dots, (q_{|Q|}, T)$, $T \leq p$. Эти пары будут состояниями нового источника W'' . За начальное состояние возьмем $(q_0, 1)$. Очевидно, дерево D_{cut} индуцирует соединение этих пар, т.е. если $b \in B$, $\beta \in B^*: \varphi(q_0, \beta) = q$, $\varphi(q, b) = q'$ и $\hat{k}(\beta, q) = k_1$, $\hat{k}(\beta b, q') = k_2$, то соединим ребром пары (q, k_1) и (q', k_2) . Соединяющему ребру припишем букву b . Повторим операцию со всеми допустимыми парами.

Таким образом, из любой пары (q, k) выходит ровно $|B|$ ребер, т.к. в дереве D_{cut} обладало этим свойством. Более того, условие $\beta \in B^*, \varphi(q_0, \beta) \in Q_F$ эквивалентно $\hat{k}(\beta, q) \geq p$, что равносильно тому, что путь соответствующий слову β ведет в источнике W'' в состояние $(\varphi(q_0, \beta), p)$. Будем считать пары вида

$(*, p)$ финальными состояниями. Получаем, что W'' представляет язык $L_p(V_{cut}) = L_p(D') = L$ посредством финальных состояний $\{(*, p)\}$.

Естественным образом, источник W'' порождает конечный автомат V'' без выхода, с выделенным множеством финальных состояний,

² Заметим, что дерево D_{cut} , вообще говоря, не задает единственным образом источник, однако достаточно для построения источника с одним дополнительным «поглощающим» состоянием, переходы из которого возможны только в него же. Цвет поглощающего состояния описывается числом, меньшим p .

совпадающим со множеством финальных состояний источника W'' . Автомат V'' представляет язык L с помощью множества финальных состояний. Покажем, что автомат V'' обладает правильной раскраской. Рассмотрим функцию k цвета состояния автомата. Пусть для состояния (q, m) автомата V' , $k((q, m)) = m$. Осталось заметить, что введенная раскраска p -правильная. ■

Доказательство Следствия 1.2:

Предположим обратное, найдется $p > 1$ и автомат W , что $L = L_p(W)$. В силу связности автомата V , найдется $b \in B$, что $\kappa_W(b) \geq p > 1$, в то время как для оставшихся букв алфавита B верно неравенство $\kappa_W(b') \leq 1$, где $b' \in B$, $b' \neq b$. Последнее неравенство выполнено, т.к. для любой буквы $b_1 \in B$ существует $b_2 \in B$, что $\kappa_W(b_1 b_2) \geq p$, в силу связности V . Но $\sum_{b \in B} \kappa_W(b) \leq |B|$, из мощностных соображений, противоречие. ■

Доказательство Теоремы 1.6:

Рассмотрим некоторый регулярный язык L над алфавитом B . Пусть $V = (B, Q, \varphi, q_0)$ представляет L с помощью множества финальных состояний $Q_F \subseteq Q$. Согласно Утверждению 1, для всех натуральных $i > |B|^{|Q|-1}$, $L \notin \mathcal{L}_i$, а согласно Теореме 1, проверка, на принадлежность L к $\mathcal{L}_i$ требует проверки $O(|Q|)$ различных автоматов. Тем самым количество автоматов, которое надо проверить алгоритму, для нахождения всех i , что $L \in \mathcal{L}_i$ оценивается сверху числом C , где $C = O(|Q||B|^{|Q|})$. ■

Глава 2. Автоматные мультимножества и распознавание с помощью гистограммной автоматной функции.

2.1 Алгоритм восстановления гистограммной автоматной функции по конечному мультимножеству.

Под мультимножеством над множеством B — будем понимать пару (B, χ) , где $\chi: B \rightarrow \mathbb{N}$ — это функция, сопоставляющая каждому элементу множества B некоторое натуральное число, называемое кратностью этого элемента.

Будем рассматривать конечные мультимножества слов длины N в алфавите $E_k = \{0, 1, \dots, k-1\}$. Под мощностью мультимножества понимается число входящих в него элементов с учетом кратностей. Мощность конечного мультимножества (B, χ) далее обозначается как $|B|$:

$$|B| = \sum_{\beta \in B} \chi(\beta).$$

Через $E_k(N)$ обозначим множество всех слов из E_k^* длины не превышающей N : $E_k(N) = \{ \beta \in E_k^* : |\beta| \leq N \}$.

Рассмотрим, натуральное число m – порог распознавания. Пусть автомат $V = (E_k, Q, E_k, \varphi, \psi, q_0)$ – конечный инициальный автомат с множеством состояний Q , автоматная функция f_V которого рассматривается на словах длины N :

$$f_V: E_k^N \rightarrow E_k^N.$$

Пусть κ_V – гистограммная функция автомата V .

Определение 2.1:

Мультимножество (B, χ) назовем автоматным, если $\kappa_V(\beta) = \chi(\beta)$, для любого $\beta \in B$, и назовём квази-автоматным, если существует автомат $V = (E_k, Q, E_k, \varphi, \psi, q_0)$, что для любого слова $\beta \in B$ $\kappa_V(\beta) \geq \chi(\beta)$. Функцию κ_V , в этом случае, назовем гистограммным покрытием.

Мы будем решать задачу проверки мультимножеств слов на автоматность и квази-автоматность. Далее мы опишем подход к задаче распознавания, основанный на свойствах гистограммной функции автомата.

Наиболее простым случаем распознавания слов автоматом V при таком подходе является проверка того, что число прообразов некоторого слова $\beta \in E_k^N$ при автоматном отображении f_V не менее некоторого порога m .

$$|\{ \alpha \in E_k^N : f_A(\alpha) = \beta \}| \geq m,$$

что эквивалентно:

$$\kappa_V(\beta) \geq m.$$

С другой стороны, нам потребуется также построить автомат V по конечному обучающему мультимножеству (B, χ) , где

$$B \subseteq E_k^*, \quad \chi: B \rightarrow \mathbb{N},$$

или проверить, что такого автомата не существует.

Согласно [2], число приведенных автоматов вида

$$(E_k, Q, E_k, \varphi, \psi, q_0)$$

при $|Q| \rightarrow \infty$ для $k > 2$ асимптотически равно

$$(nk)^{nk}/n!,$$

а для $k=2$:

$$e^{-k^2/2} \cdot (nk)^{nk} / n!$$

где $n=|Q|$. Сложность переборного решения задачи поиска автомата сверхэкспоненциально зависит от числа слов в (B, χ) , что делает переборный алгоритм не применимым на практике в случаях большого N . Тем не менее, удалось найти алгоритм, позволяющий решить указанную задачу за линейное время от мощности обучающей выборки $|B|$.

Установим ряд вспомогательных утверждений и лемм. Будем считать, что на E_k задан порядок

$$0 < 1 < 2 \dots < k-1,$$

который порождает естественный лексикографический порядок на E_k^n

$$(00 \dots 0) < (00 \dots 1) < \dots < (k-1 \ k-1 \ k-1)$$

Для натурального n , рассмотрим функцию

$$\rho_d: E_k^n \times E_k^n \rightarrow \mathbb{N},$$

определенную по правилу

$$\rho_d(\alpha, \beta) = n - \max_{\gamma \in \Gamma(\alpha, \beta)} |\gamma|,$$

где $\Gamma(\alpha, \beta)$ множество общих префиксов слов α и β .

$$\Gamma(\alpha, \beta) = \{ \gamma \mid \alpha = \gamma \alpha_1, \beta = \gamma \alpha_2 \}.$$

Функция ρ_d 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: $\rho_\partial(001, 101) = 3$, $\rho_\partial(010, 222) = 3$

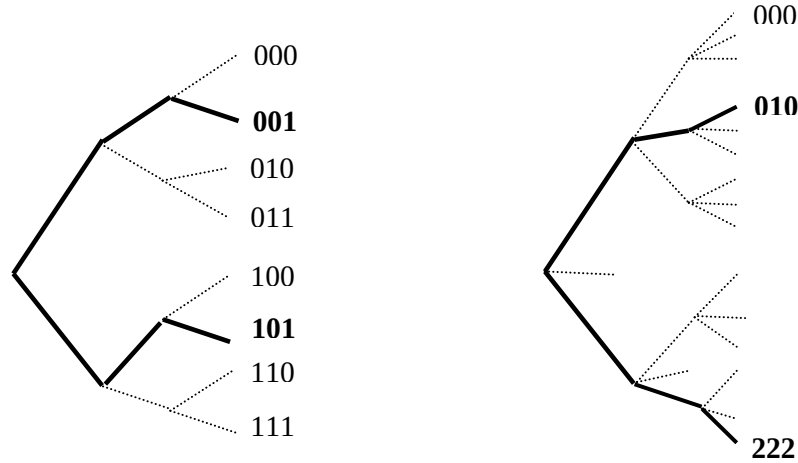


Рисунок 12. Вычисление функции ρ_∂ .

Утверждение 2.1:

Для любых трех лексикографически упорядоченных слов $\alpha \leq \gamma \leq \beta$ одинаковой длины выполнено

$$\rho_\partial(\alpha, \beta) = \max (\rho_\partial(\alpha, \gamma), \rho_\partial(\gamma, \beta)).$$

Доказательство:

Рассмотрим произвольные α, β . Пусть

$$\rho_\partial(\alpha, \beta) = r.$$

Тогда

$$\alpha = \omega \delta_1 \alpha_1, \beta = \omega \delta_2 \beta_1,$$

где

$$|\omega| = |\alpha| - r, \text{ а } \delta_1 \neq \delta_2.$$

Рассмотрим произвольное γ , лежащее между словами α и β (в лексикографическом порядке):

$$\alpha \leq \gamma \leq \beta.$$

Ясно, что

$$\gamma = \omega \delta_3 \gamma_1,$$

причем выполняется хотя бы одно из неравенств:

$$\delta_1 \neq \delta_3 \text{ или } \delta_2 \neq \delta_3.$$

Поэтому или

$$\rho_\partial(\alpha, \gamma) = r,$$

или

$$\rho_\partial(\gamma, \beta) = r,$$

или оба равенства верны. ■

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

Утверждение 2.2:

Для любых слов одинаковой длины α, β : $\alpha \leq \beta$, выполнено

$$\rho_\partial(\alpha, \beta) = \max_{\alpha \leq \gamma \leq \beta} (\rho(\alpha, \gamma), \rho(\gamma, \beta)).$$

Доказательство:

Суть, непосредственное следствие предыдущего утверждения, т.к. в качестве «промежуточного» слова можно выбирать любое слово, принадлежащее интервалу. ■

Утверждение 2.3:

Функция ρ_∂ – является метрикой на множестве E_k^n .

Доказательство:

Будем рассматривать слова одной длины, тогда для введенной функции выполнены:

$$1) \rho_\partial(\alpha, \beta) = 0 \Leftrightarrow \alpha = \beta$$

$$2) \rho_\partial(\alpha, \beta) = \rho_\partial(\beta, \alpha)$$

$$3) \rho_\partial(\alpha, \beta) \leq \rho_\partial(\alpha, \gamma) + \rho_\partial(\gamma, \beta), \text{ так как для}$$

$$\alpha \leq \gamma \leq \beta:$$

$$\rho_\partial(\alpha, \beta) \leq \max(\rho_\partial(\alpha, \gamma), \rho_\partial(\gamma, \beta)) \leq \rho_\partial(\alpha, \gamma) + \rho_\partial(\gamma, \beta).$$

из Утверждений 2.1, 2.2,

Для γ не принадлежащему этому интервалу или

$$\rho_{\partial}(\alpha, \beta) \leq \rho_{\partial}(\alpha, \gamma) \leq \rho_{\partial}(\alpha, \gamma) + \rho_{\partial}(\gamma, \beta)$$

или

$$\rho_{\partial}(\alpha, \beta) \leq \rho_{\partial}(\gamma, \beta) \leq \rho_{\partial}(\alpha, \gamma) + \rho_{\partial}(\gamma, \beta).$$

Следовательно, для каждого натурального n , ρ_{∂} – суть расстояние на множестве E_k^n . ■

Определение 2.2:

Функцию $\rho_{\partial}: E_k^n \times E_k^n \rightarrow \mathbb{N}$ назовем *расстоянием по дереву*.

Эту функцию иногда называют также лексикографической метрикой.

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

Для произвольного натурального n и произвольной словарной функции

$$f: E_k^n \rightarrow E_k^n$$

имеет место

Лемма 2.1 (о сжимающем отображении):

- 1) Функция f является детерминированной на словах длины n тогда и только тогда, когда для любых двух слов α, β одинаковой длины выполнено

$$\rho_{\partial}(\alpha, \beta) \geq \rho_{\partial}(f(\alpha), f(\beta)).$$

- 2) Функция f является детерминированной на словах длины l тогда и только тогда, когда для любых двух соседних слов α, β из E_k^l выполнено

$$\rho_{\partial}(\alpha, \beta) \geq \rho_{\partial}(f(\alpha), f(\beta)).$$

Доказательство:

- 1) Если f не является автоматной, то существуют два слова α, β с одинаковым началом длины l , а образы этих слов $f(\alpha), f(\beta)$ имеют одинаковое начало меньшей длины $m < l$. Значит

$$\rho_{\partial}(\alpha, \beta) < \rho_{\partial}(f(\alpha), f(\beta)).$$

Из детерминированности функции f следует

$$\rho_{\partial}(\alpha, \beta) \geq \rho_{\partial}(f(\alpha), f(\beta)).$$

2) Истинность прямого утверждения следует из первого пункта.

Пусть теперь для любых пар соседних слов

$$\alpha_i, \alpha_{i+1}$$

выполнено

$$\rho_{\partial}(\alpha_i, \alpha_{i+1}) \geq \rho_{\partial}(f(\alpha_i), f(\alpha_{i+1})).$$

Покажем индукцией по n , что тогда для любых пар

$$\alpha_i, \alpha_{i+n}$$

верно

$$\rho_{\partial}(\alpha_i, \alpha_{i+n}) \geq \rho_{\partial}(f(\alpha_i), f(\alpha_{i+n})).$$

Заметим, что рассматриваются слова одной длины.

База индукции ($n=1$) вытекает из формулировки.

Пусть теперь утверждение верно для n . Докажем его для $n+1$.

Обозначим попарные расстояния между словами и их образами так, как это сделано на *Рисунке 13*. По предположению индукции верно:

$$\begin{aligned} \rho_{\partial}(\alpha_i, \alpha_{i+n}) &= r_1 \geq r_1' = \rho_{\partial}(f(\alpha_i), f(\alpha_{i+n})) \\ \rho_{\partial}(\alpha_{i+n}, \alpha_{i+n+1}) &= r_2 \geq r_2' = \rho_{\partial}(f(\alpha_{i+n}), f(\alpha_{i+n+1})) \end{aligned}$$

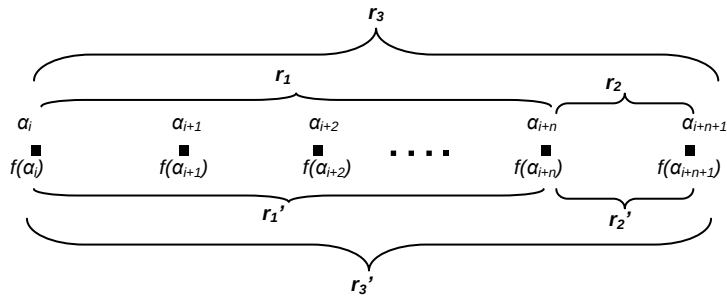


Рисунок 13. Расстояния между словами и их образами.

Нужно установить истинность:

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

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 $r_3' \leq N - (N-r_1') = r_1' \leq r_1 \leq r_3$. 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:

$$\begin{aligned} f(\alpha_i) &= \omega_1 \delta_1 \beta_1 \\ f(\alpha_{i+n}) &= \omega_1 \delta_2 \beta_2 = \omega_2 \delta_2' \beta_2' \\ f(\alpha_{i+n+1}) &= \omega_2 \delta_3 \beta_3, \end{aligned} \quad (*)$$

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 $\rho_d(f(\alpha_i), f(\alpha_{i+n+1})) \leq (N - |\omega_1|) = r_1' \leq r_1 \leq r_3$.

Аналогично для случая $r_1' < r_2'$ (если он вообще возможен:). Шаг индукции доказан. Рассмотрим теперь произвольные слова α, β ³. Ясно, что найдутся натуральные i, n , что $\alpha = \alpha_i$, $\beta = \alpha_{i+n}$, следовательно $\rho_d(\alpha, \beta) = \rho_d(\alpha_i, \alpha_{i+n}) \geq \rho_d(f(\alpha_i), f(\alpha_{i+n})) = \rho_d(f(\alpha), f(\beta))$, что доказывает ограниченную детерминированность функции f . ■

Для произвольного упорядоченного множества $B = \{\beta_1, \dots, \beta_p\}$ 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 множества $B = \{\beta_1, \dots, \beta_p\}$ и перестановки $\pi \in S_p$ обозначим через $\pi(B) = \{\beta_{\pi(1)}, \dots, \beta_{\pi(p)}\}$.

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

³ Здесь мы рассматриваем α лексикографически меньшее, чем β .

Утверждение 2.4:

Пусть B – лексикографически упорядоченное множество слов одной длины. Тогда для любой перестановки $\pi \in S_{|B|}$ выполнено:

$$R(B) \leq R(\pi(B)).$$

Доказательство:

Рассмотрим произвольное множество слов $B = \{\beta_1, \dots, \beta_p\}$ одной длины.

Из Утверждения 2.2 вытекает, что для произвольных i, j выполнено:

$$\rho_\partial(\beta_i, \beta_j) = \max_{\beta_i \leq \beta \leq \beta_j} (\rho(\beta_i, \beta), \rho(\beta, \beta_j)).$$

Таким образом, ни для какой перестановки сумма всех расстояний между соседями уменьшиться не может. ■

Определение 2.3:

Будем говорить, что автомат $V = (E_k, Q, E_k, \varphi, \psi, q_0)$ распознает множество $B \subseteq E_k^*$ с помощью гистограммной функции и порога $m > 0$, если $\kappa_V(\beta) \geq m$ для каждого $\beta \in B$.

Через B_m обозначим множество слов, имеющих не менее чем m прообразов при отображении f_V .

Построение автомата V по константе m и наперед заданному мультимножеству (B, χ) , назовем синтезом автомата по мультимножеству (B, χ) .

Ограничимся, пока, распознаванием слов одной длины N . Заметим, что не всякое мультимножество (B, χ) может быть автоматным образом E_k^N . Критерий того, что (B, χ) является автоматным образом E_k^N , описан в Теореме 2.1, которая дает также способ построения искомого автомата V .

Рассмотрим мультимножество (B, χ) , $B = \{\beta_1, \beta_2, \dots, \beta_c\} \subseteq E_k^N$.

Введем операцию проектирования мультимножеств. Через $(B, \chi)^{(1)}$ будем обозначать такое мультимножество $(B^{(1)}, \chi^{(1)})$, что

$$B^{(1)} = \{\beta \in E_k^{N-1} \mid \beta b \in B, \text{ для некоторого } b \in E_k\},$$

$$\chi^{(1)}: B^{(1)} \rightarrow \mathbb{N}, \quad \chi^{(1)}(\beta) = \sum_{\alpha \in \Gamma(\beta)} \chi(\alpha), \text{ для всех } \beta \in B^{(1)},$$

$$\Gamma(\beta) = \{\alpha \mid \exists b \in E_k: \alpha b = \beta \in B\}.$$

Введенную операцию проектирования можно применять $i > 1$ раз:

$$(B, \chi)^{(i)} = (((B, \chi)^{(1)} \dots)^{(1)}).$$

Без ограничения общности, будем считать, что слова из множества B упорядочены лексикографически и использовать знак $\leq$ для лексикографического порядка.

Следующая теорема содержит критерий возможности построения гистограммной функции автомата по мультимножеству мощности k^N .

Теорема 2.1:

Пусть (B, χ) мультимножество слов длины N над алфавитом E_k , k, N -натуральные и $\sum_{\beta \in B} \chi(\beta) = k^N$. Тогда для существования автомата

$V = (E_k, Q, E_k, \varphi, \psi, q_0)$: *такого что $\kappa_V(\beta) = \chi(\beta)$ для всех $\beta \in B$, необходимо и достаточно, чтобы для любого натурального $0 < i < N$, $(B, \chi)^{(i)}$ удовлетворяло условию:*

$$\chi^{(i)}(\beta) = c(\beta)k^i$$

для некоторого натурального $c(\beta)$.

Доказательство:

Пусть имеется мультимножество (B, χ) , $\sum_{\beta \in B} \chi(\beta) = k^N$ и автомат V , такой, что $\kappa_V(B) = \chi(B)$. Обозначим словарное отображение, задаваемое автоматом V через f_V . Отображение f_V - детерминированное, задает отображение множества с кратностями $\hat{E}_k^N$ всех префиксов слов из E_k^N во множество $\hat{B}$ - всех префиксов множества 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.

Обратно. Рассмотрим мультимножество (B, χ) , $B = \{\beta_1, \beta_2, \dots, \beta_c\} \subseteq E_k^N$.

Пусть $x_i = \chi(\beta_i)$ - кратность вхождения слова β_i и $\sum_{i=1}^c x_i = k^N$ и выполнено условие теоремы.

Сформируем мультимножество (E_k^N, χ_0) дополнив множество B до E_k^N словами нулевой кратности. Т.е. $\chi_0(\beta) = \begin{cases} \chi(\beta), & \text{если } \beta \in B \\ 0, & \text{иначе} \end{cases}$. Упорядочим слова в (E_k^N, χ_0) лексикографически. Рассмотрим слово β на позиции i в (E_k^N, χ_0) , обозначим $x_i^N = \chi_0(\beta)$. Индекс N указывает на длину слова в данном случае.

Рассмотрим представление множества слов B в виде нагруженного префиксного дерева. Каждой вершине i -го уровня приписаны слово и кратность соответствующего слова в мультимножестве $(B, \chi)^{(N-i+1)}$, $i=1, \dots, N$. Листьям же дерева припишем кратности соответствующих слов из B . Обозначим $T = k^N$.

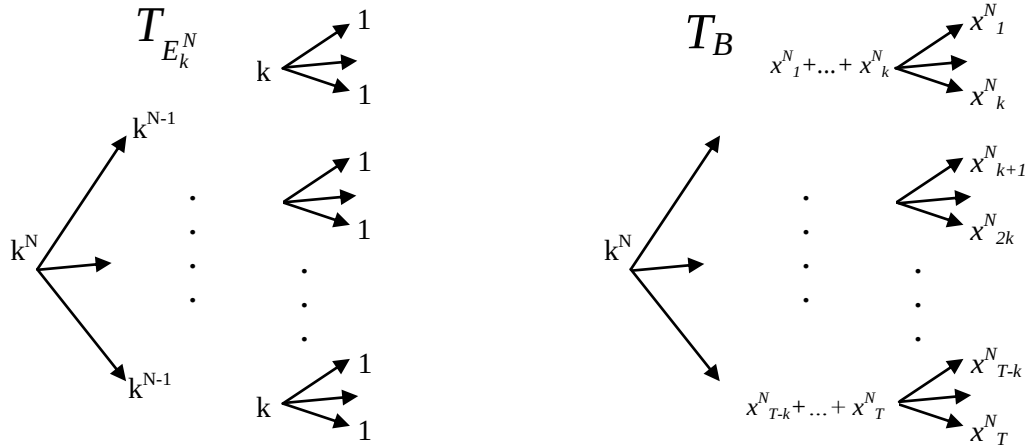


Рисунок 14. Представление множества E_k^N и мультимножества (B, χ) слов с кратностями в виде нагруженных деревьев.

Заметим, что некоторым листьям (вершинам) могут быть приписаны нули, так как соответствующее слово может вообще не входить в B .

Пусть выполнены условия теоремы. Построим искомое автоматное отображение.

Рассмотрим N -ый уровень дерева для множества B . Будем двигаться (см. рисунок 14 справа) по листьям дерева, сопоставляя первому листу дерева $T(B)$ x_1^N первых листьев дерева $T(E_k^N)$ (рисунок 14 слева), второму листу – следующие x_2^N листьев и так далее. Таким образом листьям дерева $T(B)$ сопоставлены листья дерева $T(E_k^N)$ (причем каждый лист дерева $T(E_k^N)$ 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 сопоставим вершины дерева E_k^N , из которой выходят прообразы этих листьев. Ясно, что кратности сохраняться 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 построению детерминированное и сохраняющее кратности, которое и будет задавать нам искомое автоматное отображение f . ■

Для произвольного действительного числа x , через $\lceil x \rceil$ обозначим минимальное целое, превосходящее или равное x . Суть $\lceil x \rceil$ - потолок числа x .

Пусть даны мультимножество (B, χ) и положительное r . Тогда через $r(B, \chi)$ обозначим мультимножество (B, χ_2) , где $\chi_2(\beta) = \lceil r \cdot \chi(\beta) \rceil$.

Обозначим через $E_k(N)$ множество всех слов длины не превосходящей N :

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

Также для натурального $i \leq N$, через $(B, \chi)|_i$ обозначим подмножество всех слов длины i из множества (B, χ) . Очевидно, $(B, \chi)|_i$ также является мультимножеством. Следующая теорема обобщает *Теорему 2.1* на случай когда B – суть множество слов из $E_k(N)$.

Теорема 2.2:

Пусть (B, χ) - мультимножество слов из $E_k(N)$, $\sum_{\beta \in B} \chi(\beta) = \frac{k^{n+1} - 1}{k - 1} - 1$.

Тогда для существования автомата $V = (E_k, Q, E_k, \varphi, \psi, q_0)$, такого что $\kappa_V(\beta) = \chi(\beta)$ для всех $\beta \in B$ необходимо и достаточно, чтобы для любого натурального $i < N$: $((B, \chi)|_{i+1})^{(1)} = k(B, \chi)|_i$.

Доказательство:

Пусть существует автоматная функция $f: f(E_k(N)) = B$. Тогда утверждение теоремы вытекает из детерминированности f . Обратно, пусть для всех допустимых i выполнено $(B|_{i+1})' = kB|_i$. Тогда выполнены условия *Теоремы 2.1* для множества $B|_N$. Построим отображение $f: f(E_k^N) = B|_N$. Оно и будет искомым. Действительно, $f(E_k^i) = B|_i$, для всех $0 < i < N$. ■

Вышеизложенные теоремы этого параграфа формулируют необходимые и достаточные условия для восстановления гистограммной автоматной функции, по ее части, определенной на всех словах определенной длины. Однако, в большинстве случаев, обучающая выборка задано не полностью и гистограммную функцию требуется восстанавливать по ограниченному набору слов.

Рассмотрим мультимножество (B, χ) , где B – произвольное подмножество B^* . Имеет место

Теорема 2.3:

- 1) *Существует алгоритм $\mathcal{A}$, который по мультимножеству (B, χ) , $\sum_{\beta \in B} \chi(\beta) \leq k^N$ находит автомат V , что κ_V -*

гистограммное покрытие (B, χ) , или определяет, что такого автомата не существует.

- 2) Число операций сложения и умножения натуральных чисел, необходимых для проверки и поиска не превосходит

$$C \cdot \sum_{\beta \in B} \chi(\beta).$$

- 3) При этом число $|Q_V|$ состояний автомата V удовлетворяет неравенству

$$|Q_V| \leq \sum_{\beta \in B} \chi(\beta) \cdot \frac{1 - \frac{1}{k^{N-1}}}{k - 1} + N - 1,$$

где N – максимальная длина слова из B .

Доказательство:

Рассмотрим мультимножество (B, χ) . Пусть $B = \{\beta_1, \dots, \beta_M\}$, где $\beta_i \in E_k^N$, и пусть $y_i = \chi(\beta_i)$ для всех допустимых i . Иллюстрации приведены для случая $k=2$. Заметим, что здесь и далее любое дерево подразумевается ориентированным естественным образом, т.е. ребро, соединяющее вершину q_i уровня i с вершиной q_{i+1} уровня $i+1$, суть – (q_i, q_{i+1}) .

Шаг 1:

Построим дерево $T(E_k^N)$ глубины N , и припишем каждому листу, соответствующему слову α , расстояние по дереву между α и словом, сопоставленным вышестоящему листу. Листу, соответствующему слову из одних нулей, припишем 0.

Шаг 2:

Лексикографически упорядочим слова множества $B = \{\beta_1, \dots, \beta_M\}$.

И сопоставим первым y_1 листьям, начиная с листа, соответствующего 0^N , слово β_1 (см рисунок 16).

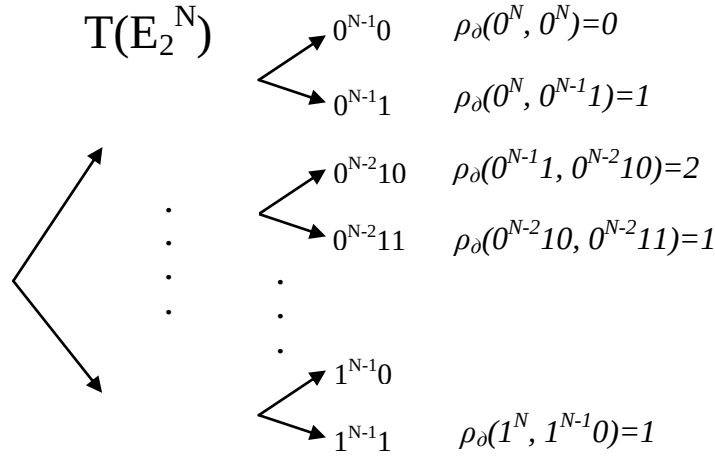


Рисунок 15. Расстояние по дереву между соседними листьями.

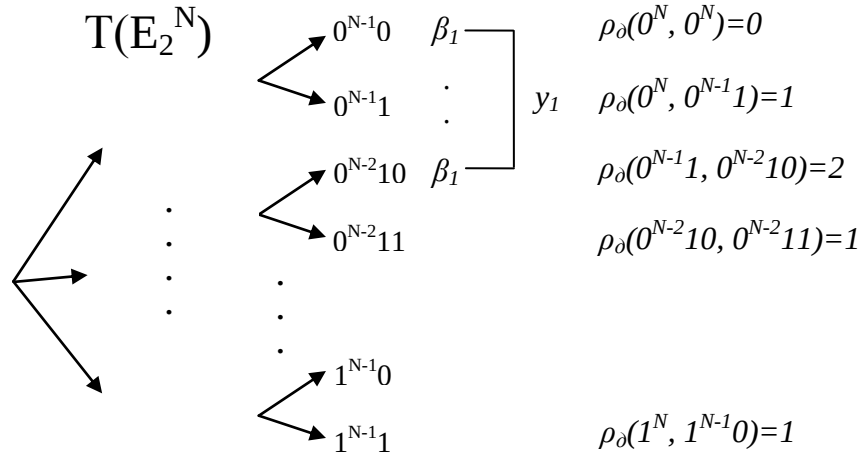


Рисунок 16. Сопоставление первых y_1 слов листьям дерева.

Шаг 3:

Пусть $\rho_\delta(\beta_{i-1}, \beta_i) = r_i$, $i \geq 2$. Будем двигаться по листьям построенного дерева сверху вниз, пока не найдем лист с приписанным расстоянием $\rho_\delta(\gamma, \delta) \geq r_i$, где δ – приписанное этому листу слово, γ – “вышестоящее” слово.

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

Замечание 1:

Если этого сделать нельзя – автомата не существует.

Замечание 2:

Если нет листа с условием $\rho_\delta(\gamma, \delta) \geq r_i$, то искомого автомата не существует.

Шаг 4:

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

Шаг 5:

Листья, которым не было сопоставлено ни одно слово, назовем пустыми. Доопределим полученную функцию в вершинах словами-соседями сверху (т.е. ближайшими словами, согласно расстоянию по дереву). Каждому листу, таким образом, сопоставлено слово из B .

Построенный N -ый уровень информационного дерева однозначно задает функцию $f: E_k^{N+1} \rightarrow B$. Пусть алгоритм $\mathcal{Q}$ завершил работу и построил некоторую функцию f . Покажем, что f – детерминированная. Действительно, по построению, для любых двух слов-соседей α_i, α_{i+1} выполнено $\rho_\delta(\alpha_i, \alpha_{i+1}) \geq \rho_\delta(f(\alpha_i), f(\alpha_{i+1}))$. Тогда по Лемме 2.1 о сжимающем отображении она является детерминированной на словах длины $i+1$. Доопределим ее по детерминированности на слова меньших длин.

Покажем, что если алгоритм $\mathcal{Q}$ обнаружил отсутствие функции f , то не существует никакой другой 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 отождествлении⁴.

⁴ Лист, соответствующий числу 0 – самый первый. Далее – сверху дерева к низу.

Докажем, что если существует некоторое детерминированное отображение f' , удовлетворяющее условиям теоремы, то обязательно будет существовать отображение f - результат применения алгоритма. Воспользуемся индукцией по длине j слов множества B .

База индукции, $j=1$. Пусть существует некоторое детерминированное отображение $f': f'(E_k)=B_1 \supseteq B$. В этом случае существует детерминированное отображение f , переводящее E_k в лексикографически упорядоченное множество $B_2 \supseteq B$. В самом деле, из Утверждения 2.4 сумма попарных расстояний в πB_1 не увеличилась, где $\pi \in S_{|B|}$: πB_1 -лексикографически упорядоченное B_1 . Пусть $f: f(E_k) = \pi B_1$, тогда, каждое слово β_i будет назначено какому-либо листу при отображении f , так как то же множество различных букв (но с не меньшей суммой расстояний между соседями) было назначено листьям при отображении f' . Любая перестановка букв, вообще говоря, не изменит детерминированности отображения ☺. Легко видеть, что полученное отображение f – суть результат применения алгоритма.

Индуктивный переход. Полагая гипотезу верной для всех $j \leq i$, установим ее истинность для $j = i+1$. Положим существование детерминированного отображения $f': f'(E_k^{i+1}) = B_1 \supseteq B$. Функция f' по детерминированности может быть доопределена до $f'(E_k^i) = B_{1,1} \supseteq B'$, где B' – есть множество префиксов длины i слов множества B . По предположению индукции, существует детерминированная функция $f: f(E_k^i) = B_2 \supseteq B'$, где f – результат применения алгоритма $\mathcal{Q}$, а B_2 - лексикографически упорядочено. Рассмотрим любое слово β_i из B' , пусть его кратность в мультимножестве $\frac{1}{k}(B, \chi)^{(1)}$ равна x_i . Обозначим

$$y_j = \begin{cases} \chi(\beta_i b_{j-1}), & \text{если } \beta_i b_{j-1} \in B \\ 0, & \text{иначе} \end{cases}. \text{ Из детерминированности } f' \text{ вытекает: } \sum_{i=1}^k y_i \leq kx_i.$$

Это неравенство позволяет продолжить информационное дерево, соответствующее f , на один уровень, получив, в итоге, искомое отображение. На информационном дереве, соответствующем f , x_i листьям сопоставлено слово β_i . Все слова B , начинающиеся на β_i можно, сохраняя порядок, приписать листьям с учетом кратностей, причем эти листья имеют общее ребро с теми вершинами, которым приписано слово β_i , сверх того, индуцированное отображение при таком построении будет детерминированным. Повторяя построение для всех слов β_t , $t \neq i$, предъявим отображение $f : f(E_k^{i+1}) \supseteq B$. По построению, f является результатом применения алгоритма $\mathcal{A}$. Первое утверждение теоремы доказано.

Стоит заметить, что финальный 5 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 близости к обучающей выборке, или из соображений минимальности получаемого автомата.

Изменим шаг 5 алгоритма для достижения оценок из 3 пункта теоремы. Пусть для фиксированного B , алгоритм $\mathcal{A}$ доработал до 5 шага и построил ориентированное дерево T_1 . Каждую вершину, из которой существует путь в непустой лист обозначим меткой ω (отметим, что согласно определению пути в ориентированном графе, полученный путь соединяет вершину i уровня с некоторой вершиной $(i+1)$ -го уровня дерева и других путей нет). Образует из T_1 новое дерево T_2 , вычеркивая ребра, конец которых совпадает с непомеченной вершиной. Очевидно, что достроив и доопределив листья дерева T_2 произвольным образом, можно построить ограниченно - детерминированное отображение, удовлетворяющее пункту 1. В частности, по дереву T_2 можно построить

автомат с $|Q_2| = q_2 + 1$ состояниями, где одно и то же состояние приписывается каждой вычеркнутой вершине, а q_2 совпадает с числом вершин дерева T_2 , нагруженных меткой ω .

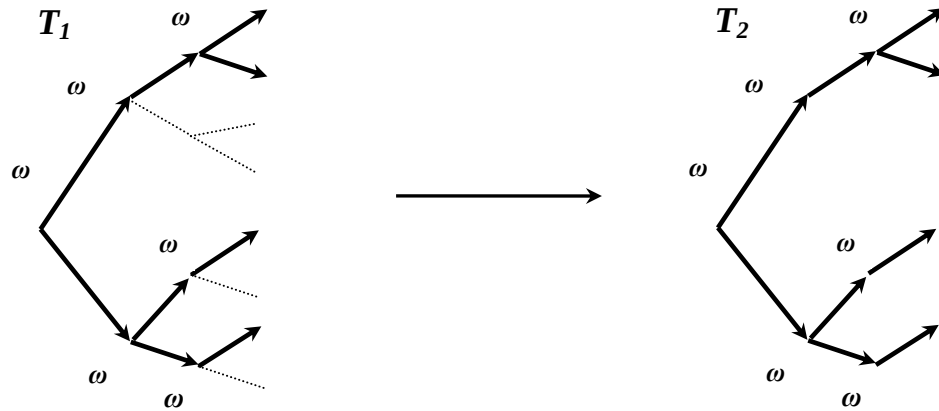


Рисунок 17. Построение дерева T_2 .

Пусть по дереву T_2 получено частично определенное отображение f_2' . То есть тем вершинам, которым приписана метка ω , сопоставлено какое-либо состояние отображения. Доопределим частично-определенное f_2' до детерминированного f_2 , приписав каждой непомеченной ω вершине одно и то же состояние. Дерево T_2 преобразуем к дереву T_3 следующим образом: если из некоторой вершины q_1 ведет только одно ребро, скажем, в вершину q_2 , то вычеркнем его, совместив вершины q_1 и q_2 , приписав совмещенной вершине (q_1, q_2) .

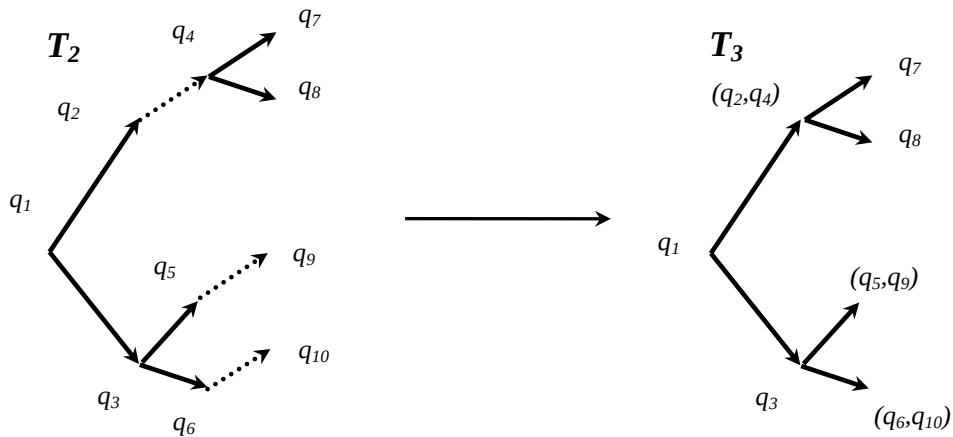


Рисунок 18. Построение дерева T_3 .

Отображение f_2 задает некоторый автомат $V_2=(A, Q_2, B, \varphi_2, \psi_2, q_2^0)$, $Q_2 = \{q_1, \dots, q_n\}$, удовлетворяющий условиям теоремы. Используя структурное соединение с автоматом счетчиком, образуем автомат V_3 , также удовлетворяющий условию теоремы, но обладающий меньшим числом состояний. Для этого заметим, что любое состояние $q \in Q_2$ либо приписано некоторой вершине дерева T_3 , либо принадлежит некоторому набору $\bar{q} = (q_{i_1}, \dots, q_{i_s})$, приписанному одной из вершин дерева T_3 . Определим функции переходов и выходов автомата V_3 следующим образом:

$$\varphi_3(\bar{q}_i, t, a) = \begin{cases} \varphi_2(q_i, a), & \text{если } \bar{q}_i = (q_i) \\ \bar{q}_i, & \text{если } \bar{q} = (q_{i_1}, \dots, q_{i_s}), t < s \\ \varphi_2(q_{i_s}, a) & \text{если } \bar{q} = (q_{i_1}, \dots, q_{i_s}), t = s \end{cases}$$

$$\varphi_3(q_i, t, a) = \varphi_3(\bar{q}, t, a), \quad q_i \in \bar{q}$$

$$\psi_3(\bar{q}_i, t, a) = \begin{cases} \psi_2(q_i, a), & \text{если } \bar{q}_i = (q_i) \\ \psi_2(q_{i_s}, a), & \text{если } \bar{q} = (q_{i_1}, \dots, q_{i_s}) \end{cases}$$

Где t – текущее состояние счетчика из N состояний.

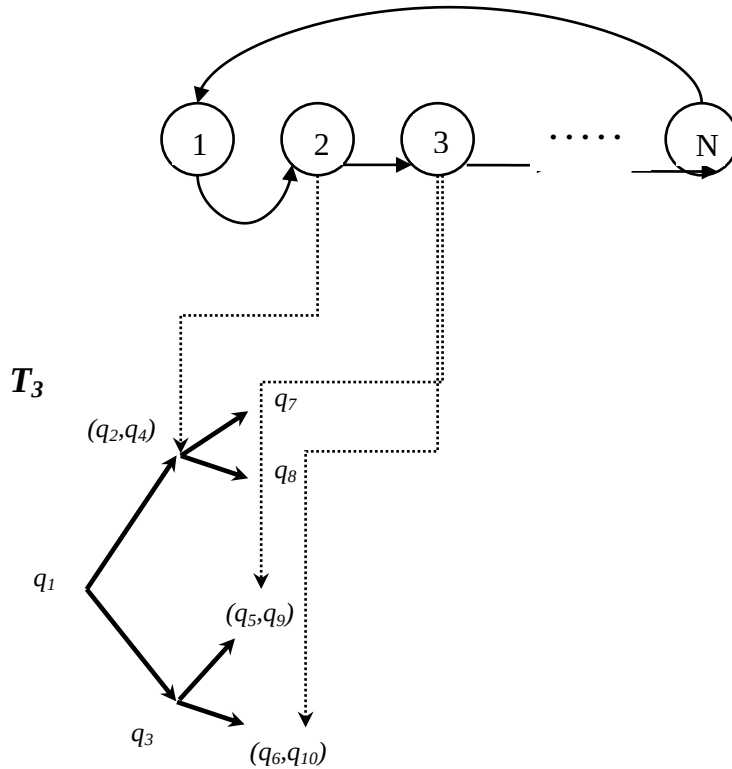


Рисунок 19. Построение дерева T_3 .

Таким образом, $Q_3 = Q_3' \cup Q_3'' \cup Q_3^C$.

$$Q_3' = \{q \in Q_2 \mid q \text{ приписан вершине } T_3\},$$

$$Q_3'' = \{\bar{q} \mid \bar{q} \text{ приписан вершине } T_3\},$$

$$Q_3^C = \{t_1, \dots, t_N\} - \text{состояния счетчика.}$$

Из построения ясно, что $V_3 = (A, Q_3, B, \varphi_3, \psi_3, q_2^0)$ имеет на выходе то же множество, что и V_2 . Однако число состояний в $Q_3' \cup Q_3''$ не превосходит

число вершин дерева T_3 , то есть $|Q_3'| + |Q_3''| < |B| \cdot \frac{1 - \frac{1}{k^{N-1}}}{k-1}$, откуда прямо следует оценка из условия теоремы. Что и требовалось. ■

Отметим, что сложность синтезированного автомата здесь зависит линейно от мощности обучающей выборки. Следующая *Теорема 2.4* оценивает сложность подсчета значения гисторгаммной функции на слове. Имеет место

Теорема 2.4:

*Пусть $V = (A, Q, B, \varphi, \psi, q_0)$, κ_V – его гистогаммная функция. Тогда для любого слова β выходного алфавита B , число C арифметических операций для нахождения $\kappa_V(\beta)$ удовлетворяет $C = |B| * |\beta| * |Q|$.*

Доказательство:

Пусть $V = (A, Q, B, \varphi, \psi, q_0)$. Сопоставим каждой букве b выходного алфавита квадратную целочисленную матрицу $M_b = \{m_{ij}\}$ размера $|Q| \times |Q|$, $0 \leq m_{ij} \leq N$, 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 B^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 $|B|$ ненулевых элементов, и эти матрицы фиксированы получаем утверждение теоремы. ■

2.2 О классах L_p - языков.

Если для некоторых $j \leq i$, $L \in \mathcal{L}_i$ будет выполнено $L \in \mathcal{L}_j$, то отсюда не следует, что для классов языков выполнено $\mathcal{L}_i \subseteq \mathcal{L}_j$. Оказывается, верна

Теорема 2.5:

Для любых натуральных $i \neq j$ выполнено: $\mathcal{L}_i \not\subseteq \mathcal{L}_j$.

Доказательство:

Пусть даны натуральные $i < j = i + k$ для некоторого k . Покажем, что $\mathcal{L}_i \not\subseteq \mathcal{L}_j$. Пусть $E_p = \{a_1, \dots, a_p\}$, где $p > 1$. Покажем, что существует язык $L' \in \mathcal{L}_i$: $L' \notin \mathcal{L}_j$. Для этого предъявим конечный детерминированный автомат $V = (E_p, Q_V, E_p, \varphi_V, \psi_V, q_V)$, что $L' = L_i(V)$, и для любого автомата W вида $W = (E_p, Q_W, E_p, \varphi_W, \psi_W, q_W)$: $L' \neq L_j(W)$.

Через f_V обозначим выходную функцию автомата V , и пусть $L = L_i(V)$. Рассмотрим подмножество слов длины n в L : $L|_n = \{\alpha \in L \mid |\alpha| = n\}$. Пусть теперь множество $L|_n = \{\alpha_1, \dots, \alpha_s\}$ состоит из подряд идущих слов, начиная с $\alpha_1 = a_1 \dots a_1$, где $E_p = \{a_1, \dots, a_p\}$, очевидно, такое существует для некоторого s . Здесь $s = s(n)$ может принимать конечное число значений S . Выберем s максимальным среди всех возможных из S .

Рассмотрим поддереву $T(E_p^n)$ глубины n информационного дерева, соответствующего автомату V . Так как f_V детерминированная функция и $L|_n(V) = \{\alpha \in f_V(E_p^n) \mid \kappa_V(\alpha) \geq i\}$, то каждому листу дерева $T(E_p^n)$ приписано

либо слово из $L|_n$, либо слово кратности меньшей i (других нет по построению). Пусть η_t - число вершин дерева, которым не приписано ни одно слово из $L|_n$, но вершине сверху и вершине снизу приписаны α_t и α_{t+1} соответственно.

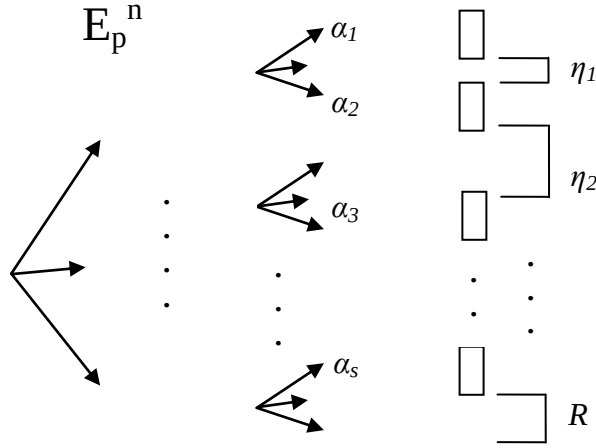


Рис. 20. Поддерево автомата V с приписанными выходными словами $\alpha_1, \dots, \alpha_s$.

Оценим $s(n)$ для произвольного n . Из того, что $|L|_n| \leq p^n$, вытекает $s(n) \leq \left\lceil \frac{p^n}{i} \right\rceil$. Покажем, что найдется натуральное $n = n(p, i)$, что верхняя оценка

достигается: $s(n) = \left\lceil \frac{p^n}{i} \right\rceil$ для любых p, i . Для этого заметим, что набор

расстояний S , соответствующий мультимножеству $L|_n$, обладает свойством: любое натуральное j , которое встречается в этом наборе, встречается только на позиции, кратной p^{j-1} . Этот факт вытекает из следующего рассуждения. Если слова α_t, α_{t+1} из $L|_n$ таковы, что $\rho_d(\alpha_t, \alpha_{t+1}) = j$, то $t = rp^{j-1}$, по построению $L|_n$, где r - некоторое натуральное. Следовательно, данное j может встречаться в наборе S только на позиции с номером irp^{j-1} . Поэтому, если $L|_n = \{\alpha_1, \dots, \alpha_s\}$, где $s = s(n)$ - максимальное число различных наборов, то набор S , соответствующий $L|_n$, будет меньше или равен набору R_n , согласно Утверждению 2.3. Следовательно, применяя к набору $\underline{S}$ алгоритм из Теоремы 2.3, мы получим что для $L|_n$

существует детерминированная словарная функция $f: E_p(n) \rightarrow E_p(n)$, что $\kappa_f(L)=n$. Итак, $s(n(p,i)) = \left\lfloor \frac{p^{n(p,i)}}{i} \right\rfloor$, для любых p,i .

Для доказательства теоремы достаточно показать, что для любых $i < j$, найдется натуральное m : $\left\lfloor \frac{p^m}{i} \right\rfloor > \left\lfloor \frac{p^m}{j} \right\rfloor$. Действительно, из неравенства

будет следовать существование языка $L = \{\alpha_1, \dots, \alpha_w\}$, где $w = \left\lfloor \frac{p^m}{i} \right\rfloor$, что

$L \in \mathcal{L}$, но $L \notin \mathcal{L}'$. Пользуясь неравенством $\left\lfloor \frac{p^m}{i} \right\rfloor = \frac{p^m}{i} - \left\{ \frac{p^m}{i} \right\} > \frac{p^m}{i} - 1$, покажем

более сильное утверждение: для любых i, j найдется m :

$$\frac{p^m}{i} - 1 > \frac{p^m}{j}, \text{ откуда будет следовать } \left\lfloor \frac{p^m}{i} \right\rfloor > \frac{p^m}{i} - 1 > \frac{p^m}{j} > \left\lfloor \frac{p^m}{j} \right\rfloor.$$

$$p^m(j-i) > ij, \text{ что, очевидно, выполнено для } m > \left\lceil \log_p \left(\frac{ij}{j-i} \right) \right\rceil + 1.$$

Покажем, теперь, что $\mathcal{L}' \subsetneq \mathcal{L}$. Пусть, как и ранее, даны натуральные $i < j = i + k$ для некоторого k . Покажем, что найдется язык L' , что $L' \in \mathcal{L}'$, но $L' \notin \mathcal{L}$. Для этого предъявим конечную L часть языка L' , которая обладает этим свойством. Используя детерминированность, по конечной части L всегда можно будет восстановить язык L' , возможно, не единственным образом. Пусть язык L составлен из некоторого числа s подряд идущих слов длины n . Пусть L является подмножеством языка $L' \in \mathcal{L}'$, $L = \{\alpha_1, \dots, \alpha_s\}$. Тогда существует автоматная функция $f: E_p^* \rightarrow E_p^*$, и, соответствующая ей, вторая автоматная функция $\kappa_f: E_p^* \rightarrow \mathbb{N} \cup \{0\}$, что $\kappa_f(\alpha_t) \geq j$, $t=1, \dots, s$. Обозначим через F дерево, соответствующее ограниченно-детерминированной функции f . Ясно, что n -ому уровню дерева приписано не менее js слов. Следовательно,

$(n-1)$ -ому уровню дерева F , приписано $\left\lceil \frac{j \cdot s}{p} \right\rceil$ слов, где $\lceil x \rceil$ - потолок числа

x , т.е. минимальное целое большее или равное числу x . Вообще, из

детерминированности вытекает, что на каждом следующем уровне будет $\left\lceil \frac{y}{p} \right\rceil$ слов, где y – число слов на текущем уровне дерева. Обобщая это правило, составим таблицу

Уровень №	n	(n-1)	...	(n-m)	...	1
Минимальное число слов	js	$\left\lceil \frac{j \cdot s}{p} \right\rceil$	...	l_f	...	1

Табл. 1. Минимальное число слов на уровнях дерева F , если $\kappa_f(\alpha_j) \geq j$.

Предположим, теперь, что $L' \in \mathcal{L}$. Тогда найдется автоматная функция $g: E_p^* \rightarrow E_p^*$, и, соответствующая ей, вторая автоматная функция $\kappa_g: E_p^* \rightarrow \mathbb{N} \cup \{0\}$, что $\kappa_g(\alpha_t) \geq i$, $t=1, \dots, s$. Обозначим через G дерево, соответствующее ограниченно-детерминированной функции g . Ясно, что n -ому уровню дерева приписано не менее is слов. Аналогично построим таблицу и для дерева G .

Уровень №	n	(n-1)	...	(n-m)	...	1
Минимальное число слов	is	$\left\lceil \frac{i \cdot s}{p} \right\rceil$	...	l_g	...	1

Табл. 2. Минимальное число слов на уровнях дерева G , если $\kappa_g(\alpha_j) \geq i$.

Покажем теперь найдутся числа s, m, n , что $l_g = i$, но $l_f = j - 1$. Это будет выполнено, если (система возникает из рассмотрения Табл. 1):

$$\begin{cases} s \leq p^m \cdot (1 - \frac{1}{j}) \\ s > (1 - \frac{2}{j}) \cdot p^m - \frac{\sum_{t=1}^{m-1} p^t}{j} \end{cases}$$

и (система возникает из рассмотрения Табл. 2):

$$\begin{cases} s \leq p^m \\ s > (1 - \frac{1}{i}) \cdot p^m - \frac{\sum_{t=1}^{m-1} p^t}{i} \end{cases}$$

Добавим условие существования искомого набора из s подряд идущих слов для обеих функций f, g :

$$\begin{cases} s \cdot i \leq p^n \\ s \cdot j \leq p^n \end{cases}$$

Выразим условие, что на $(n-m)$ -ом уровне достаточно прообразов, чтобы построить детерминированные отображения f и g .

$$\begin{cases} i \leq p^{n-m} \\ j-1 \leq p^{n-m} \end{cases}$$

Будем требовать, чтобы все s подряд идущих слов из набора, имели одинаковое начало длины $n-m$:

$$s \leq p^m.$$

Откуда, используя $i=j-k > j-1 > 0$, комбинируя системы и удалив зависимые неравенства, получаем систему:

$$\begin{cases} s \leq p^m \cdot (1 - \frac{1}{j}) \\ s > (1 - \frac{2}{j}) \cdot p^m - \frac{\sum_{t=1}^{m-1} p^t}{j} \\ s > (1 - \frac{1}{i}) \cdot p^m - \frac{\sum_{t=1}^{m-1} p^t}{i} \\ s \leq p^m \\ s \leq \frac{p^n}{j} \\ m \leq n - \log_p(j-1) \end{cases}$$

Положим $s = \left\lfloor p^m \cdot (1 - \frac{1}{j}) \right\rfloor$, где $[x]$ – целая часть числа x . Тогда первое и

четвертое неравенства, очевидно, выполнены. Пусть теперь

$m = \lfloor n - \log_p(j-1) \rfloor$, тогда будут выполнены неравенства 6 и 5. Далее

заметим, что неравенство $\frac{\sum_{t=1}^{m-1} p^t}{j} > 1$ влечет за собой $\frac{\sum_{t=1}^{m-1} p^t}{i} > 1$, и т.к.

$s = \left\lceil p^m \cdot \left(1 - \frac{1}{j}\right) \right\rceil > \left(1 - \frac{1}{j}\right) \cdot p^m - 1 > \left(1 - \frac{1}{i}\right) \cdot p^m - 1$, то при условии $\frac{\sum_{t=1}^{m-1} p^t}{j} > 1$ будут

выполнены второе и третье неравенства системы.

$$\frac{\sum_{t=1}^{m-1} p^t}{j} = \frac{1}{j} \cdot \frac{p^m - 1}{p - 1} > 1, \Leftrightarrow p^m - 1 > j(p-1), \Leftrightarrow m > \log_p(j(p-1)-1), \text{ учитывая}$$

ограничения на m , получаем $[n - \log_p(j-1)] > \log_p(j(p-1)-1)$, что истинно,

$$\text{если } n-1 - \log_p(j-1) > \log_p(j(p-1)-1), \Leftrightarrow \frac{p^{n-1}}{j-1} > j(p-1) - 1 \Leftrightarrow p^{n-1} > (j^2 - j)(p-1) -$$

$$j+1 \Leftrightarrow p^{n-1} > j^2 p - j^2 - jp + 1. \text{ Что эквивалентно}$$

$$j^2(p-1) - jp + 1 - p^{n-1} < 0.$$

$$D = p^2 + 4(p-1)(p^{n-1} - 1) = 4p^n - 4p^{n-1} + p^2 - 4p + 4 \geq 0, \text{ при } p \geq 1, j = \frac{p \pm \sqrt{D}}{2(p-1)}. \text{ При } p \geq 1,$$

$$D \geq p^2, \text{ поэтому } \frac{p - \sqrt{D}}{2(p-1)} \leq 0 \text{ таким образом, неравенство будет выполнено}$$

$$\text{при } j > \left\lceil 1, \frac{p + \sqrt{p^2 + 4(p^{n-1} - 1)(p-1)}}{2(p-1)} \right\rceil. \text{ Заметим, что всегда варьируя длину}$$

слова n , можно подобрать длину интервала нужным образом.

Тем самым всегда будет существовать слово α^0 длины $n-m$, которое сопоставлено ровно $j-1$ вершинам уровня $n-m$ дерева F , т.е. $\kappa_f(\alpha^0) = j-1$, но сопоставлено i вершинам уровня $n-m$ дерева G . По построению, α^0 – суть общее начало слов $\alpha_1, \dots, \alpha_s$. Это доказывает существование языка L , что $L \in \mathcal{L}$, но $L \notin \mathcal{L}_j$, $j > i$ ■

Глава 3. Распознавание через синтез.

3.1 Распознавание слов автоматами.

Пусть задан конечный алфавит A и два множества

$$\bar{A}_1, \bar{A}_2 \subseteq A^*$$

слов из алфавита A , которые являются обучающими выборками для бесконечных множеств

$$U_1, U_2 \subseteq A^*$$

соответственно. При распознавании слова $\alpha \in A^*$, необходимо отнести к U_1 или к U_2 , некоторым способом.

Наиболее известным способом распознавания слов автоматом

$$V = (A, Q, B, \varphi, \psi, q_0),$$

Является представимость слов в автомате. Слово $\alpha \in A^*$ представимо в автомате V с выделенным множеством выходных букв $B_1 \subseteq B$, точно тогда, когда последний символ выходного слова

$$\psi(q_0, \alpha) \in B_1 \subseteq B.$$

Согласно теорема Клини [1], класс распознаваемых таким образом слов совпадает с классом регулярных языков.

На практике наиболее распространен алгоритм распознавания слов с помощью синтеза цепей Маркова [10] и Скрытых Марковских Моделей [8]. Здесь используется распознавание через синтез. Создается генератор слов, выдающий слова так часто как это наблюдается в обучающих выборках. Зачастую, генератор выдает слова не реже, чем в обучающей выборке. Обычно строится два вероятностных источника λ_i ($i=1,2$) по двум обучающим выборкам $\tilde{A}_i$ ($i=1,2$). Решение принимается по методу максимального правдоподобия и заключается в сравнении вероятностей $P(\alpha | \lambda_1)$, $P(\alpha | \lambda_2)$: если больше первая вероятность, относим слово $\alpha \in A^*$ к U_1 , в противном случае – к U_2 . Основной сложностью здесь является построение СММ λ_i по обучающей выборке V_i . Практика показывает, что с этой задачей успешно справляется алгоритм Баума-Уэлша [5]. Алгоритм рекурсивно строит вероятностный источник с наибольшей вероятностью появления на выходе слов из обучающей выборки. Обзор использования вероятностных источников в задаче распознавания речи приведен в [11].

Под вероятностным источником (СММ) принято понимать пятерку

$$\lambda = (S, A, F, G, \pi)$$

где

$$S = \{s_1, s_2, \dots, s_N\}$$

- совокупность состояний источника,

$$A = \{a_1, a_2, \dots, a_M\}$$

- алфавит наблюдаемой последовательности,

$$F = \{f_{ij}\}$$

- стохастическая матрица вероятностей переходов между состояниями,

$$f_{ij}=P[q_{t+1}=s_j | q_t=s_i],$$

$$G=\{g_{ij}\}$$

- стохастическая матрица вероятностей появления символов

$$g_{jk}=g_j(k)=P[a_k | q_t=s_j].$$

$\pi=(\pi_1, \pi_2, \dots, \pi_N)$ - вектор вероятностей начального состояния

$$\pi_i=P(q_0=S_i), \quad 1 \leq i, j, k \leq N.$$

Рассмотрим слово $\alpha=a_1...a_T$ длины T . Алгоритм Баума-Уэлша обучения вероятностных источников использует оценку функции правдоподобия $P(\alpha|F,G,\pi)$. Для эффективного подсчета используется прямая и обратная функции частичного правдоподобия для $0 < t < T$:

$$\mu_t(i)=P(a_1...a_t | q_t=i), \quad \nu_t(j)=P(a_{t+1}...a_T, | q_t=j).$$

«Прямая» функция $\mu_t(i)$ – это условная вероятность того, что к моменту времени t выдано слово $a_1...a_t$ и в этот же момент времени вероятностный источник находится в состоянии под номером i . «Обратная» функция $\nu_t(j)$ есть условная вероятность наблюдения слова $a_{t+1}...a_T$, начиная с момента времени $t+1$, при условии, что источник находился в состоянии с номером j . Если инициализировать начальные значения введенных функций естественным образом:

$$\mu_1(i)=\pi_i g_i(a_1), \quad \nu_T(j)=1, \quad i,j=1,...|Q|,$$

то можно заметить, что прямая и обратная функции правдоподобия допускают рекурсивный счет:

$$\mu_{t+1}(i)=g_i(a_{t+1}) \sum_{j=1}^{|Q|} \mu_t(j) f_{ji}, \quad \nu_t(i)=\sum_{j=1}^{|Q|} \nu_{t+1}(j) f_{ij} g_j(a_{t+1}), \quad t=T-1, \dots, 1, \quad 0 \leq i \leq |Q|.$$

Верно равенство:

$$P(v|F,G,\pi)=\sum_{i=1}^{|Q|} \mu_T(i)$$

Таким образом, вычисление вероятности по формуле требует выполнения $O(|Q|T^2)$ операций, что заметно меньше, чем требует прямой счет $O(|Q|^T T)$. С помощью введенных функций можно обучать

вероятностные источники чаще выдавать слово $\alpha = a_1 \dots a_T$. Если заменить матрицы A, B в вероятностном источнике на новые матрицы $\hat{A}, \hat{B}$, полученные по нижеприведенным формулам (3), то получим

$$P(\alpha | \hat{F}, \hat{G}) \geq P(\alpha | F, G)$$

и, более того, итеративное применение формул

$$\hat{a}_{ij} = \frac{\sum_{t=1}^T \mu_{t-1}(i) f_{ij} g_j(a_t) v_t(j)}{\sum_{t=0}^{T-1} \mu_t(i) v_t(i)}, \quad \hat{b}_{jk} = \frac{\sum_{t=1, a(t)=k}^T \mu_t(i) v_t(j)}{\sum_{t=0}^T \mu_t(i) v_t(i)} \quad (3)$$

где $1 \leq i, j \leq |S|$, $1 \leq k \leq |A|$, гарантирует сходимость к оценкам максимального правдоподобия параметров F, G , то есть к локальному максимуму функции $f(F, G) = P(v | F, G)$ [8].

Несмотря на широкое распространение в инженерной среде, Скрытые марковские модели продолжают оставаться объектом не до конца изученным с математической точки зрения. Одна из основных проблем СММ - проблема остановки обучения. При длительном итеративном применении формул обучения, источник «настраивается» на обучающую выборку.

Обычно обучение останавливает учитель. Он же выбирает начальные параметры скрытой марковской модели, которую будут обучать. Зачастую, это делается исходя из знания физической сущности задачи. Необходимо, также, заметить, что в общем случае, при применении обучающего алгоритма, нет никакой гарантии, что обученный вероятностный источник будет доставлять глобальный максимум функции правдоподобия. Учитель выбирает несколько параметрически близких друг к другу модели и обучает все на одной и той же выборке. Впоследствии, проводится эксперимент, определяющий наиболее подходящий вероятностный источник. Такой подход существенно замедляет процесс обучения модели.

Наконец, распознавание с помощью вероятностных источников вычислительно трудоемко. Методы распознавания предполагают подсчет значения функции правдоподобия $P(o_1, o_2, \dots, o_T | \lambda)$, что требует порядка $|S|T^2$ операций умножения и сложения действительных чисел, где $|S|$ – число состояний модели. Если длина распознаваемого слова и множество возможных СММ (например, как в распознавании речи) велики, а решение нужно принимать в реальном времени, подсчет $P(o_1, o_2, \dots, o_T | \lambda)$ может оказаться неприемлем. В конечном счете, не исследован вопрос о том, какие языки могут быть распознаны вероятностными источниками. Эти факты убедили автора, что алгоритм распознавания, свободный от вышеперечисленных проблем будет интересен. В следующем параграфе представлены результаты эксперимента такого метода, основанного на синтезе гистограммных функций детерминированного автомата.

Рассмотрим вероятностный источник заданный в виде пятерки $\lambda = (S, A, F, G, \pi)$. Для произвольного неотрицательного числа ε , через $\Gamma(\varepsilon)$ обозначим множество слов $\{ \gamma \in A^* \mid P(\gamma | \lambda) \geq \varepsilon \}$. Таким образом, $\Gamma(\varepsilon)$ – множество выходных слов вероятностного источника λ , появляющихся на его выходе с вероятностью большей или равной заданного действительного числа. Оказывается, что с точностью до конечной добавки, множество $\Gamma(\varepsilon)$ является множеством начал периодических последовательностей из некоторого конечного множества периодических сверхслов. Сформулируем теорему о структуре множества $\Gamma(\varepsilon)$:

Теорема 3.1:

Для произвольных $\lambda = (S, A, F, G, \pi)$ и $\varepsilon > 0$ выполнено одно из условий

1) Либо $|\Gamma(\varepsilon)| < \infty$.

2) Либо $\Gamma(\varepsilon) = \{ \alpha \beta^\infty \}_n \mid \alpha \in \{ \alpha_1, \dots, \alpha_R \}, \beta \in \{ \beta_1, \dots, \beta_T \}, n$ – произвольное натуральное }, где R, T – некоторые натуральные числа.

Доказательство:

Пусть заданы СММ $\lambda=(S, A, F, G, \pi)$, $\varepsilon > 0$ и $\Gamma(\varepsilon)$. Для каждого выходного слова $\alpha=a_1a_2...a_L$, $L < 2^m + 2$, определим рекурсивно набор действительных положительных величин $\xi_i(\alpha)$:

- 1) $\xi_i(a)=f_{i1}g_1(a)+ f_{i2}g_2(a) + ... + f_{in}g_n(a)$, для каждой буквы a и состояния s_i .
- 2) $\xi_i(\alpha a)=f_{i1}g_1(a)\xi_i(\alpha)+ f_{i2}g_2(a)\xi_i(\alpha) ... + f_{in}g_n(a)\xi_i(\alpha)$, для произвольного слова $\alpha a \in A^*$.

Легко видеть что $\xi_i(\alpha)$ – есть вероятность наблюдать на выходе слово α , с условием нахождения модели λ в начальный момент времени в состоянии s_i . Более точно

$$\xi_i(\alpha) = P(o_t...o_{t+L}=\alpha \mid q_{t-1}=s_i), \text{ где } L=|\alpha|.$$

Ясно, что $P(\alpha a \mid \lambda) = P(\alpha a) = P(\alpha, q_L=s_1) \cdot (f_{11}g_1(a) + f_{12}g_2(a) + ... + f_{1n}g_n(a)) + ... + P(\alpha, q_L=s_n) \cdot (f_{n1}g_1(a) + f_{n2}g_2(a) ... + f_{nn}g_n(a)) = \sum_{i=1}^n P(\alpha, q_L = s_i) \cdot \xi_i(\alpha)$ где $a \in A$, $\alpha \in A^*$, $L=|\alpha|$.

Аналогично доказывается, что для $\alpha_1, \alpha_2 \in A^*$: $P(\alpha_1 \alpha_2 \mid \lambda) = P(\alpha_1 \alpha_2) = \sum_{i=1}^n P(\alpha_1, q_M = s_i) \cdot \xi_i(\alpha_2)$, где $M=|\alpha_1|$.

Если для всех слов $\alpha_1, \alpha_2, ..., \alpha_{m^L}$ длины $L=2^n+1$ выходного алфавита выполнено $\xi_i(\alpha_j) < 1 \ \forall i$, то $\Gamma(\varepsilon)$ – конечно. Действительно, пусть $\xi(\alpha) = \max_{i=1,...,n} \xi_i(\alpha)$ для каждого слова $\alpha_1, \alpha_2, ..., \alpha_{m^L}$, тогда

$$P(\alpha \alpha') = \sum_{i=1}^n P(\alpha, q_L = s_i) \cdot \xi_i(\alpha') \leq \xi(\alpha') \sum_{i=1}^n P(\alpha, q_L = s_i) = \xi(\alpha') \cdot P(\alpha) \leq \xi_{\max} \cdot P(\alpha)$$

для любых слов $\alpha \in A^*$ и $\alpha' \in A^L$, где $\xi_{\max} = \max_{\alpha \in A^L} (\max_{i=1,...,n} \xi_i(\alpha)) = \max_{\alpha \in A^L} \xi(\alpha)$.

Пользуясь тем, что $\xi_{\max} < 1$, получаем, что с увеличением длины слова на L , вероятность наблюдать данное слово на выходе модели уменьшается не менее, чем в $1/\xi_{\max} > 1$ раз. Тем самым при увеличении длины слова, вероятность его выдачи источником стремится к 0 и только

конечное множество слов $\gamma \in V^*$ может удовлетворять $P(\gamma|\lambda) \geq \varepsilon$, для произвольного $\varepsilon > 0$.

Пусть теперь множество $A_1 = \{ \alpha \in A^L \mid \xi(\alpha) = 1 \}$ непусто, пусть также $A_2 = A^L \setminus A_1$. Обозначим $\xi_{\min} = \max_{\alpha \in A_2} (\max_{i=1, \dots, n} \xi_i(\alpha)) = \max_{\alpha \in A_2} \xi(\alpha)$. Из неравенства $\xi_{\min} < 1$ следует существование натурального r : $(\xi_{\min})^r < \varepsilon$. Из неравенства $P(\alpha' \alpha'') \leq P(\alpha') \xi_{\min} \leq \xi_{\min}$, где $\alpha'' \in A_2$, вытекает, что если некоторое выходное слово α содержит более r , где $r < \log(\varepsilon) / \log(\xi_{\min})$, слов из A_2 , то $\alpha \notin \Gamma(\varepsilon)$. Предположим, что множество $\Gamma(\varepsilon)$ бесконечно. Следовательно, оно содержит слова сколь угодно большой длины. Поэтому почти все слова (за исключением конечной добавки) $\alpha \in \Gamma(\varepsilon)$ содержат подслова $\alpha' \in A_1$. Каждому $\alpha' \in A_1$ соответствует цепочка подмножеств состояний $Q_1, \dots, Q_L$. $Q_1 \subseteq S$ – состояния в которые попадает модель, находясь в предыдущий момент времени в состоянии s_i , важно заметить, что выход вероятностного источника λ при переходе из s_i в Q_1 – буква a_1 . $Q_{i+1} \subseteq S$ – состояния модели в которые она попадает из Q_i . В силу $\xi(\alpha') = 1$, для каждого $1 \leq i \leq L$: $P(q_{t+1} \in Q_{i+1} \mid q_t \in Q_i) = 1$, т.е. из состояний Q_i модель может перейти только в состояния множества Q_{i+1} . Так как $L > 2^{|S|}$, то среди $Q_1, \dots, Q_L$ найдется пара одинаковых $Q_k = Q_l$, для некоторых натуральных $k \neq l$ (можно считать, что k, l – минимальные натуральные). Это означает, что для любого допустимого j : $Q_{k+j} = Q_{l+j}$ и $a_{k+j} = a_{l+j}$, и слово $\alpha' = a_1 a_2 \dots a_L = a_1 a_2 \dots a_{k-1} \tilde{\alpha}^d \tilde{\alpha}'$, где d – натуральное, а $\tilde{\alpha}'$ – начало $\tilde{\alpha}$. Таким образом любое слово из A_1 является началом некоторого периодического сверхслова. Более того, если $\alpha_1 = \gamma \alpha \zeta$, $\alpha_2 = \gamma \alpha \zeta'$, где $\gamma, \zeta, \zeta' \in A^*$, $\zeta \neq \Lambda$, $\zeta' \neq \Lambda$, то $\zeta = \zeta'$ и $\alpha_2 = \alpha_1$. Действительно, пусть $r < (l - k + 1)$ – остаток от деления $(L - l)$ на $(l - k + 1)$: $L = Z(L - l)(l - k + 1) + r$, Z – некоторое целое, тогда переход из Q_L определяется однозначно (с вероятностью 1): $Q_L = Q_{l+r} \rightarrow Q_{l+rem}$, где rem – остаток от деления $(r + 1)$ на $(l - k + 1)$ и, стало быть, следующая за α буква слова α_1 и α_2 определяется однозначно. Аналогично и последующие

буквы, следовательно $\alpha_1 = \alpha_2$. Из однозначности переходов вытекает также, что $\zeta = \tilde{v}'' \tilde{\alpha}^d \tilde{v}'''$, где $\tilde{v}' \tilde{v}'' = \tilde{\alpha}$, $\tilde{v}'$ - начало $\tilde{\alpha}$ ■.

Роль вероятностного источника в задаче распознавания заключается в приписывании словам вероятностей. Другим способом приписывания словам весов является использование гистограммной автоматной функции, описанное в главе 2. В следующем параграфе приводится пример использования такого подхода в задаче классификации электрокардиограмм.

3.2 Пример использования гистограммной автоматной функции для распознавания слов.

Для демонстрации применимости метода распознавания, основанного на гистограммных функциях автомата, был проведен эксперимент по отличению электрокардиограмм здоровых людей, от электрокардиограмм пациентов с нарушениями проводимости сердца. Нарушение проводимости состояло в полной АВ блокаде в сочетании с блокадой левой ножки пучка Гиса у человека. Вообще, для изучения возможностей метода были выбраны электрокардиограммы, т.к. сигнал ЭКГ от сердечных сокращений обладает регулярной структурой [9], что соотносится с детерминированной природой распознавателя. Разность потенциалов, снимаемых электрокардиографом с поверхности тела человека, принято разделять на Р-волну QRS-комплекс и Т-волну по названиям локальных экстремумов на графике. Изменение порядка локальных экстремумов невозможно [9], однако возможны сдвиг, изменение амплитуды или исчезновение некоторых локальных экстремумов, в связи с ненормальной электрической активностью сердца.

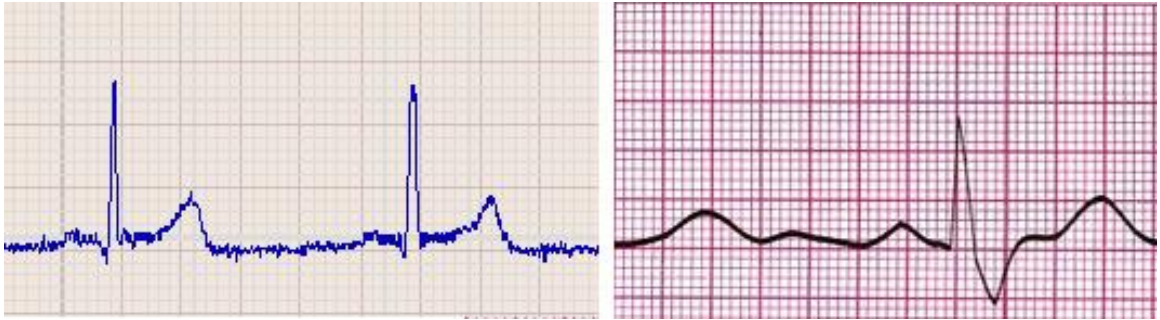


Рисунок 21. Типичные ЭКГ здорового человека (слева) и человека с нарушением проводимости сердца.

Данные для распознавания и обучения были получены от ФГБУ РКНПК Института Клинической Кардиологии им. А.Л.Мясникова. Полный корпус данных состоял из 100 ЭКГ здоровых испытуемых и 100 ЭКГ испытуемых с полной АВ-блокадой, в сочетании с в сочетании с блокадой левой ножки пучка Гиса. Для анализа использовалось отведение V5. Корпус данных был разделен на обучающую и проверочную выборки в соотношении 1:2. Вообще говоря, ЭКГ были различны по длине, в силу специфики методики снятия ЭКГ и индивидуальных особенностей испытуемого. Например, от сердца к сердцу варьируется ширина PQRS – комплекса, от силы контакта электродов с поверхностью тела варьируется и несущая частота сигнала ЭКГ, т.н. изолиния. Работа мышц (например дыхательных) также имеет влияние на чистоту снимаемого сигнала. Окончательно, встроенная в аппарат ЭКГ шумоочистка, призванная подавлять действие шумов может вносить изменения в сигнал. Метод распознавания, основанный на синтезе гистограммной автоматной функции учитывает эти ограничения.

Представим ЭКГ как вектор действительных чисел $\xi = (e_1, e_2, \dots, e_n)$ длины n . Для обучения и классификации использовались нормализованные по длине ЭКГ: ξ заменялся на вектор $\zeta = (e_{t-m}, \dots, e_t, \dots, e_{t+m}) = (f_1, \dots, f_{2m+1})$, где $e_t = \max_i \{e_i\}$. Натуральное t выбиралось исходя из

тех соображений, чтобы вектор ζ содержал все локальные экстремумы PQIRST. Например, $m = 7$.

Одним из основных плюсов метода распознавания основанного на гистограммных функциях является простота классификатора. Для минимизации числа состояний автомата, каждая ЭКГ подвергалась предварительному «огрублению» с помощью набора порогов $Th = \{t_0, \dots, t_R\}$, $R \in \mathbb{N}$, $t_{i+1} > t_i > 0$ для всех допустимых i . Представим ЭКГ ζ как вектор действительных чисел $\zeta = (f_1, f_2, \dots, f_{2m+1})$, тогда для обучения и классификации ЭКГ преобразовывалась к вектору $\zeta' = (f_1', f_2', \dots, f_{2m+1}')$, где $f_j' = \min_i \{t_i \mid t_i \geq f_j\}$. Таким образом, распознавание и обучение осуществлялись на словах алфавита E_R . Отметим, что при таком подходе, различные исходные векторы ζ_1, ζ_2 могут перейти в одно слово ζ' .

При изменении множества порогов Th , изменяется точность огрубления. Как следствие, изменяется число состояний синтезированного автомата. Автор варьировал как пороги Th , так и длину слов $2m+1$ в обучающей выборке, для достижения компромисса между точностью распознавания и сложностью синтеза автомата.

Итак, на входе алгоритма синтеза гистограммных функций имеется два мультимножества (D_1, χ_1) , (D_2, χ_2) . $D_1 = \{\zeta_1, \dots, \zeta_P\}$ – слова, соответствующие нормальным ЭКГ, $\chi_1(\zeta_i)$ – кратность вхождения слова ζ_i . Аналогично – для ЭКГ, соответствующих нарушениям сердечной проводимости.

Используя алгоритм построения гистограммного накрытия из Теоремы 3.3, автор восстановил κ_i – накрытие (D_i, χ_i) , $i=1,2$. Дальнейшая классификация слова β из проверочной выборки осуществлялась по правилу:

$$\begin{aligned} \kappa_2(\beta) \geq \kappa_1(\beta) &, \text{ то } \beta - \text{ЭКГ, соответствующая нарушению проводимости,} \\ \kappa_2(\beta) < \kappa_1(\beta) &, \text{ то } \beta - \text{ЭКГ-норма.} \end{aligned}$$

эта доля уменьшалась, что повлекло увеличение числа правильно классифицированных слов.

Полученные результаты вселяют оптимизм, т.к. с помощью детерминированного объекта удалось добиться приемлемой точности распознавания столь вариабельного объекта. Следует отметить вычислительную простоту как метода синтеза автомата, так и классификации слов.

Согласно Теореме 2.4, для вычисления значения $k_V(\beta)$ достаточно $|\beta||Q||B|$ операций. Учитывая линейную зависимость числа состояний автомата от площади обучающей выборки, получаем линейную зависимость сложности распознавания от длины слова и площади обучающей выборки.

Литература:

- [1] Kleene S.C., *Representation of events in nerve nets and finite automata*,// Automata studies, princeton university press,1956, P.3-41.
- [2] Post E., *Two-Valued iterative Systems of Mathematical Logic* // Princeton Univ. Press, Princeton, 1941
- [3] Бабин Д.Н., *О полноте двухместных о.д.-функций относительно суперпозиции* // Дискретная математика, том 1, 1989, Вып. 4, стр. 86-91
- [4] Кудрявцев В.Б., Алешин С.В., Подколзин А.С., *Введение в теорию автоматов* // Изд-во Наука, М., 1985, 320 с.
- [5] Бухараев Р.Г., *Основы теории вероятностных автоматов* // Изд-во Наука, М., 1985, 268 с.
- [6] Baum L.E., Egon J.A., *An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology* // Bull. Amer. Meteorol. Soc. Vol. 73. 1967. P.360–363.
- [7] Brandenburg F.-J., *Uniformly growing k-th power free homomorphisms* // Theor. Comp. Sci., 1983, V.23, №1, P.69-82.
- [8] Левинсон С.Е., *Структурные методы автоматического распознавания речи* // ТИИЭР, № 11, 1985, с. 100-126.
- [9] Мурашко В.В., Струтынский А.В., *Электрокардиография* // Изд-во МЕДпресс-информ, М., 2007, 320 с.
- [10] Марков А.А., *Избранные труды* // Изд-во Академии Наук СССР, 1951, Серия «Классики Науки».
- [11] 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, 2004. с.45-70.
- [12] Летичевский А.А., *Условия полноты для конечных автоматов* // Вычислительная математика и математическая физика, №4, 1961, с. 702-710.
- [13] Кратко М.И., *Алгоритмическая неразрешимость проблемы распознавания полноты для конечных автоматов* // ДАН СССР, 1964, т.155, N 1, с.35–37.
- [14] Маккалок У.С., Питтс У., *Логическое исчисление идей, относящихся к нервной деятельности* // Автоматы / Сборник статей под редакцией

Шеннона К.Э., Маккарти Дж.,-М., Изд-во иностранной литературы, 1956, С.362-385.

[15] Яблонский С.В., *Функциональные построения в k -значной логике* // Труды математического института им. В.А. Стеклова, АН СССР, 1958, Т.51, С. 5-142

[16] Алешин С.В., *Об одном следствии теоремы Крона-Роудза* // Дискретная математика, том 11, вып.4, 1999 год, С. 101- 109

Публикации автора по теме диссертации

[17] Пархоменко Д.В., *Порожденные автоматами p -языки* // Дискретная математика, 2014, Т.26, вып. 1, с. 96-102.

[18] Пархомнеко Д.В., *Вторая автоматная функция и с нею связанные классы регулярных языков* // Интеллектуальные системы, Т.17, вып. 1-4, 2013,

[19] Пархомнеко Д.В., *Метод распознавания множества слов через синтез детерминированного автомата* // Интеллектуальные системы, Т.15, вып. 1-4, 2011, с. 189-204.

[20] Пархомнеко Д.В., *Моделирование вероятностными источниками* // Интеллектуальные системы, Т.14, вып. 1-4, 2010, с. 213-224.

[21] Пархоменко Д.В., *Спектральная автоматная функция и связанные с нею регулярные языки* // Интеллектуальные системы в производстве, 2012, вып. 1, с. 165-175.

[22] Пархоменко Д. В., *Применение метода Марковских моделей к задаче описания морганий человека*, // Интеллектуальные системы в производстве, 2007, вып. 2, с. 54-61.

[23] Пархоменко Д.В., *Применение вероятностных источников к распознаванию семейств графиков* // Тезисы докладов секции «Математика и механика» Международной конференции студентов, аспирантов и молодых учёных «Ломоносов–2009», -М.: Механико-математический факультет МГУ им. М.В.Ломоносова, 2009, с. 104

[24] Пархоменко Д.В., *Регулярность частотных языков* // Материалы XVII Международной конференции «Проблемы теоретической кибернетики», Казань: Отечество, 2014, 307 с.

[25] Пархоменко Д.В., *Гистограммная функция автомата и связанные с нею классы языков* // Материалы XI Международного семинара «Дискретная математика и ее приложения», Москва, Изд-во механико-математического факультета МГУ, 2012, 453 с.

[26] Пархоменко Д.В., *Особенности моделирования графиков вероятностными автоматами* // Материалы X Международного семинара «Дискретная математика и ее приложения», Москва, Изд-во механико-математического факультета МГУ, 2010, 549 с.