Лекция №15 (продолжение)

08.11.24

IV. Регулярные языки и конечные автоматы

1. Алфавит, слово, язык

Алфавит – это произвольное непустое конечное множество $V = \{a_1, ..., a_n\}$.

Его элементы называются буквами или символами.

Слово в алфавите $V = \{a_1, ..., a_n\}$ - любой кортеж $x = (a_{i_1}, ..., a_{i_k}) \in V^k, k \geq 0$.

Пустой кортеж в данном контексте называют пустым словом и обозначают буквой λ.

Слова принято записывать без скобок и запятых просто как цепочки следующих друг за другом букв: $x = a_{i_1} a_{i_2} ... a_{i_k}$. Используется также и такая запись слова $x = x(1)x(2)...x(k)$, где $x(i)$ обозначает i-ю букву слова $1 \leq i \leq k$. Например, если $x = abca$ - слово в алфавите $\{a, b, c\}$, то $x(1) = x(4) = a, x(2) = b, x(3) = c$. Такая запись удобна тем, что позволяет не использовать двойных индексов для номеров букв в слове, а также для букв использовать то же имя (с номером в скобках), что и для всего слова. Конечно, такая «побуквенная» запись слова $x = x(1)x(2)...x(k)$ употребляется, как правило, для непустых слов, но если в ней допускается $k = 0$, то при $k = 0$ считается, что $x = \lambda$.

Синонимом термина «слово» служит термин «цепочка».

Следует подчеркнуть, что в определениях слова и, дальше, языка в алфавите **ссылка на алфавит обязательна! Нельзя говорить просто о словах и языках безотносительно заданного алфавита**, но можно подразумевать алфавит по умолчанию, если он фиксирован в определенном контексте[1].

Число букв в слове называется его длиной и обозначается $|x|$. То есть $|x| = k \rightleftharpoons x \in V^k, k \geq 0$. Понятно, что длина пустого слова равна нулю. Каждая буква алфавита есть слово длины 1 (однобуквенное слово).

Пустое слово не есть буква алфавита!

Множество всех слов в алфавите $V = \{a_1, ..., a_n\}$ обозначается V^*, а множество всех непустых слов обозначается V^+. То есть $V^* = \bigcup_{k=0}^{\infty} V^k; V^+ = \bigcup_{k=1}^{\infty} V^k = V^* \setminus \{\lambda\}$. Эти множества бесконечны, но счетны. Можно задать их нумерацию. Одна из таких нумераций, называемая лексикографической, рассмотрена в Учебнике, п. 7.1.

Равенство слов понимается как равенство кортежей, то есть

---

[1] В некоторых руководствах используют термин «слово (язык) *над* алфавитом V ». В англоязычной литературе принят термин “word (string) *over* alphabet V ”

$x = y \rightleftharpoons (|x| = |y|) \,\&\, (\forall i = \overline{1,|x|})(x(i) = y(i))$.

Проще говоря, слова одинаковой длины считаются равными, если они соответственно побуквенно совпадают. Такое равенство слов в некоторых руководствах называют графическим.

На множестве $V*$ всех слов в алфавите V определяется операция соединения (конкатенации, катенации), а именно, если $x = x(1)x(2)...x(k), y = y(1)y(2)...y(m)$, то слово

$$x \cdot y = x(1)x(2)...x(k)y(1)y(2)...y(m) \in V^{k+m}$$

называют соединением слов x и y. Эту операцию обозначают точкой, которая, как правило опускается.

Нетрудно доказать, что операция соединения ассоциативна, а пустое слово является относительно нее нейтральным элементом. Разумеется, соединение не коммутативно.

Например, $abca \cdot cab = abcacab \neq cab \cdot abca = cababca$.

Длина соединения равна сумме длин соединяемых слов.

Можно тогда определить алгебру $(V*, \cdot, \lambda)$ - множество всех слов в алфавите V с операцией соединения и пустым словом как нейтральным элементом. Ясно, что это моноид. Его называют *свободным моноидом над алфавитом* V (или *моноидом, порожденным алфавитом* V).

Замечание. Термин «свободный» связан с понятием свободных алгебр, обсуждение которого выходит за пределы нашего курса. Упрощая, можно сказать, что мы не «связываем» слова каким-либо отношением эквивалентности, кроме графического равенства слов.

В качестве примера такого отношения можно привести следующий. Пусть исходный алфавит разделен на две равные части, где каждой букве $a_i \in \{a_1, ..., a_n\}$ из первой части сопоставляется ее «двойник» $\overline{a}_i \in \{\overline{a}_1, ..., \overline{a}_n\}$ из второй части, и принимается по определению такое «равенство» (точнее, эквивалентность): $a_i\overline{a}_i = \overline{a}_i a_i = \lambda$. Два слова считаются равными, если они различаются разве лишь вхождениями таких пар букв, которые можно заменить пустым словом. Например, будут в этом смысле равными слова $ab\overline{b}\overline{a}ca = a\overline{a}ca = ca = ca\overline{c}c = c\overline{a}b\overline{b}aa$ (можно как сокращать, так и вставлять такие пары).

Можно показать, что на множестве таких слов естественно определяется структура группы[2].

Следующее понятие – основное в этом разделе.

Язык в алфавите V - любое подмножество множества всех слов в этом алфавите: $L \subseteq V*$.

Множество всех слов тогда называется *универсальным языком в алфавите* V .

---

[2] Можно определить такое отображение $h: V \to V$, что $h(a) = \overline{a}, h(\overline{a}) = a$, и распространить его на множество слов согласно формулам $h(xy) = h(x)h(y), h(\lambda) = \lambda$ (такое отображение называется морфизмом – см. Учебник, дополнение Д7.3). Тогда легко показать, что слово x^{-1}, обратное слову x, то есть такое, что $xx^{-1} = x^{-1}x = \lambda$, определяется формулой $x^{-1} = h(x^R)$, где $x^R = x(k)...x(2)x(1)$ - инверсия слова x. Например, $(\overline{a}b\overline{c}a)^{-1} = \overline{a}c\overline{b}a$.

На множестве $2^{V^*} = \{L : L \subseteq V^*\}$ всех языков в алфавите V (то есть на булеане множества всех слов), помимо теоретико-множественных операций, можно определить операцию соединения языков:

$$L_1 L_2 \rightleftharpoons \{xy : x \in L_1, y \in L_2\}.$$

Для конечных языков результат применения этой операции легко найти. Например,

$$L_1 = \{ab, acba, bab\}, L_2 = \{cbca, abab\},$$
$$L_1 L_2 = \{abcbca, ababab, acbacbca, acbaabab, babcbca, bababab\}$$

Это можно рассматривать как простое раскрытие скобок в формальном выражении

$$(ab + acba + bab)(cbca + abab) = abcbca + ababab + acbacbca + acbaabab + babcbca + bababab$$

Из этого примера видно, что операция не коммутативна. Действительно, в соединении $L_2 L_1$ есть слова, начинающиеся буквой “c”, тогда в первом соединении таких слов нет.

Определим алгебру:

$$\mathbf{L}_V = (2^{V^*}, \cup, \cdot, \varnothing, \{\lambda\})$$

Теорема. Алгебра $\mathbf{L}_V$ является замкнутым полукольцом.

Доказательство. См. Учебник, теорема 7.1.

Здесь выпишем аксиомы полукольца применительно к этому частному случаю:

1) $K \cup (L \cup M) = (K \cup L) \cup M$
2) $K \cup L = L \cup K$
3) $L \cup \varnothing = L$
4) $L \cup L = L$
5) $K(LM) = (KL)M$
6) $L \cdot \{\lambda\} = \{\lambda\} \cdot L = L$
7) $K(L \cup M) = KL \cup KM; (L \cup M)K = LK \cup MK$
8) $L \cdot \varnothing = \varnothing \cdot L = \varnothing$

Тождества (1)-(4) выражают известные свойства операции объединения множеств.

Тождество (5) является прямым следствием свойства ассоциативности операции соединения слов; тождества (6) также очевидно. Тождества (7) дистрибутивности соединения относительно объединения легко доказываются методом двух включений, а тождество (8) можно считать принятым по определению.

Так проверяется выполнение аксиом идемпотентного полукольца.

Чтобы доказать замкнутость полукольца $\mathbf{L}_V$, нужно доказать, что для любой последовательности языков $\sup L_n = \bigcup_{n=0}^{\infty} L_n$, что было сделано при рассмотрении индуктивно упорядоченных множеств (см. лекцию №5), а также доказать тождества «бесконечной дистрибутивности» (непрерывности операции соединения):

$$K\cdot\bigcup_{n=0}^{\infty}L_n=\bigcup_{n=0}^{\infty}KL_n;(\bigcup_{n=0}^{\infty}L_n)K=\bigcup_{n=0}^{\infty}L_nK\,.$$

Эти тождества без особого труда также доказываются методом двух включений по аналогии с доказательством тождеств (7).

Набросок доказательства тождеств (7)

$$x\in K(L\cup M)\Rightarrow x=yz,y\in K,z\in L\cup M\Rightarrow y\in K,z\in L\vee z\in M\Rightarrow$$
$$\Rightarrow yz\in KL\vee yz\in KM\Rightarrow x=yz\in KL\cup KM$$

Обратное включение:

$$x\in KL\cup KM\Rightarrow x\in KL\vee x\in KM\Rightarrow(x=yz,y\in K,z\in L)\vee$$
$$\vee x=yz,y\in K,z\in M)\Rightarrow(y\in K)\,\&\,((z\in L)\vee(z\in M))\Rightarrow$$
$$\Rightarrow(y\in K)\,\&\,(z\in L\cup M)\Rightarrow yz=x\in K(L\cup M).$$

Второе тождество доказывается аналогично.

Непрерывность соединения:

$$x\in K\bigcup_{n=0}^{\infty}L_n\Rightarrow x=yz,y\in K,z\in\bigcup_{n=0}^{\infty}L_n\Rightarrow y\in K,(\exists n)(z\in L_n)\Rightarrow$$
$$(\exists n)(y\in K,z\in L_n)\Rightarrow(\exists n)(x=yz\in KL_n)\Rightarrow yz\in\bigcup_{n=0}^{\infty}KL_n.$$

Обратное включение рекомендуется доказать самостоятельно. Непрерывность соединения справа доказывается аналогично.

Итак, **мы имеем замкнутое полукольцо: полукольцо всех языков в заданном алфавите**. Носитель этого полукольца является бесконечным, но уже не счетным множеством. Оно, как можно доказать, эквивалентно (равномощно) множеству всех действительных чисел.

Лекция №16

15.11.24

2. Регулярные языки и регулярные выражения

Определение регулярного языка

На множестве всех языков в алфавите V определим подмножество *регулярных* языков. Определение дается по индукции:

1) Языки пустой ($\varnothing$), состоящий из одного пустого слова ($\{\lambda\}$) и состоящий из одной однобуквенной цепочки $a\in V$ считаются регулярными.
2) Если P и Q - регулярные языки, то по определению регулярны их объединение $P\cup Q$ и их соединение PQ.
3) Если язык P регулярен, то – по определению – регулярна его итерация $P*$.
4) Никаких других регулярных языков в алфавите V не существует.

Иначе говоря, множество регулярных языков (в алфавите V, произвольно фиксированном) есть наименьшее по включению множество, содержащее элементарные регулярные языки из п. (1) записанного выше определения и замкнутое относительно объединения, соединения и итерации.

Из определения следует, что *любой конечный язык регулярен* (почему?).

Таким образом, возникает полукольцо

$$\mathbf{R}_V = (\mathrm{R}(V), \cup, \cdot, \varnothing, \{\lambda\}),$$

носителем которого является множество $\mathrm{R}(V)$ всех регулярных языков в алфавите V. В силу определения оно является подполукольцом полукольца всех языков в данном алфавите.

Однако это полукольцо не является замкнутым. Это следует из такого примера.

Рассмотрим язык $L = \{a^n b^n : n \geq 0\}$ в алфавите $V = \{a, b\}$. В конце этого раздела мы докажем, что такой язык нерегулярен. Но его можно представить в виде бесконечного объединения

$\{a^n b^n : n \geq 0\} = \bigcup_{n=0}^{\infty} \{a^n b^n\}$ последовательности регулярных языков, в которой каждый член содержит единственное слово $a^n b^n$ при фиксированном $n \geq 0$. Это значит, что не любая бесконечная последовательность регулярных языков имеет регулярную точную верхнюю грань.

Полукольцо $\mathbf{R}_V$ является так называемым *полукольцом с итерацией*. Так определяется полукольцо, являющееся подполукольцом некоторого замкнутого полукольца (в данном случае это полукольцо всех языков) и вместе с каждым своим элементом содержащим его итерацию. Дальше мы увидим, что многие результаты, касающиеся замкнутых полуколец, распространяются и на полукольца с итерацией. Прежде всего это касается размеченных над полукольцами орграфов.

Отметим в связи с этим такой результат.

Определим язык $L^{+k} = \bigcup_{n=k>0}^{\infty} L^n$ - объединение всех положительных степеней языка L,

начиная с k-й. При k=1 получаем позитивную итерацию языка L, обозначаемую просто L^+.

Теорема. Если язык L регулярен, то и язык L^{+k} регулярен при любом положительном k.

Доказательство. Записанное выше бесконечное объединение можно представить так:

$$L^{+k} = \bigcup_{n=k>0}^{\infty} L^n = \bigcup_{n=0}^{\infty} L^k L^n .$$

В силу непрерывности операции соединения получим:

$$L^{+k} = \bigcup_{n=0}^{\infty} L^k L^n = L^k \bigcup_{n=0}^{\infty} L^n = L^k L^*,$$

Откуда и следует регулярность языка L^{+k} как соединения двух регулярных языков (легко видеть, что также $L^{+k} = L^* L^k$).

Следствие. Позитивная итерация регулярного языка регулярна.

Замечание. Не следует думать, что позитивная итерация получается из итерации выбрасыванием пустого слова. Это будет верно, если сам язык не содержит пустого слова.

В общем случае $L^+ = LL^* = L^*L = \begin{cases} L^*, \lambda \in L \\ L^* \setminus \{\lambda\}, \lambda \notin L \end{cases}$.

В частности, множество всех слов в заданном алфавите есть не что иное, как итерация алфавита как языка, являющегося конечным множеством однобуквенных слов. А множество всех непустых слов будет позитивной итерацией алфавита (пустое слово не принадлежит алфавиту!).

Регулярные выражения

Для упрощения и большей наглядности записи алгебраических операций над регулярными языками вводят специальные *регулярные выражения*.

Регулярное выражение (в алфавите V) – это формула, обозначающая некоторый язык в этом алфавите. Система обозначений строится по индукции согласно определению регулярных языков. Будем использовать значок $\mapsto$ вместо слова «обозначает»: А $\mapsto$ Б читается как «А обозначает Б».

Тогда:

1) $\varnothing \mapsto \varnothing, \lambda \mapsto \{\lambda\}, a \mapsto \{a\}, a \in V$ (обозначения элементарных регулярных языков).
2) Если $\alpha \mapsto P, \beta \mapsto Q$, то $(\alpha + \beta) \mapsto P \cup Q, (\alpha\beta) \mapsto PQ$.
3) Если $\alpha \mapsto P$, то $\alpha^* \mapsto P^*, \alpha^+ \mapsto P^+$.
4) Никаких других обозначений регулярных языков посредством регулярных выражений не существует.

Можно экономить скобки, договорившись о приоритетах операций. Самый высокий приоритет имеет итерация, потом соединение и, наконец, объединение (+).

Заметим также, что при использовании символа λ возникает некоторый конфликт обозначений. Символ λ обозначает и само пустое слово и, как регулярное выражение, язык, состоящий из одного пустого слова. Из контекста каждый раз будет понятно, что именно обозначает λ.

Два регулярных выражения α и β называются тождественными, если они обозначают один и тот же язык. В этом случае формула $\alpha = \beta$ называется (регулярным) тождеством.

Основные тождества – это аксиомы полукольца, записанные посредством регулярных выражений:

1) $\alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma$
2) $\alpha + \beta = \beta + \alpha$
3) $\alpha + \varnothing = \alpha$
4) $\alpha + \alpha = \alpha$
5) $\alpha(\beta\gamma) = (\alpha\beta)\gamma$
6) $\alpha\lambda = \lambda\alpha = \alpha$
7) $\alpha(\beta + \gamma) = \alpha\beta + \alpha\gamma, (\beta + \gamma)\alpha = \beta\alpha + \gamma\alpha$

8) $\alpha \cdot \varnothing = \varnothing \cdot \alpha = \varnothing$

Из этих тождеств, а также из свойств полукольца с итерацией, можно вывести следующие тождества:

9) $\alpha\alpha^* = \alpha^*\alpha = \alpha^+$
10) $\alpha^+ + \lambda = \alpha^*$
11) $(\alpha^*)^* = (\alpha^+)^* = (\alpha^*)^+ = \alpha^*$
12) $(\alpha + \beta)^* = (\alpha^*\beta^*)^*$
13) $\varnothing^* = \lambda$
14) $\alpha^+ + \alpha^* = \alpha^*$

Тождество (12) особенно интересно. Его можно назвать аналогом бинома Ньютона для полуколец с итерацией. Его легко обосновать, если α и β понимать как буквы некоторого алфавита. Тогда левая часть тождества обозначает множество всех слов в алфавите $\{\alpha, \beta\}$, а каждое слово в этом алфавите можно описать так: идет какая-то цепочка букв α, возможно пустая, потом какая-то цепочка букв β, также возможно, что пустая; и это чередование цепочек этих двух букв происходит сколько угодно раз, а может быть, ни разу (тогда получается пустая цепочка). Используя запись множества через коллективизирующее свойство, это можно выразить так:

$(\alpha + \beta)^* = \{\alpha^{m_1}\beta^{n_1}\alpha^{m_2}\beta^{n_2}...\alpha^{m_k}\beta^{n_k} : m_i, n_i \geq 0, 1 \leq i \leq k, k \geq 0\}$.

Можно обобщить этот результат для произвольных регулярных выражений α и β.

Отсюда видно также, насколько обозначение регулярного языка в виде регулярного выражения проще и компактнее теоретико-множественного описания.

Рассмотрим по этому поводу еще такой пример.

Пусть язык L определен регулярным выражением

$L = (a^* + (bc)^+)^*$.

Преобразуем его согласно тождеству (12), а также (11):

$(a^* + (bc)^+)^* = ((a^*)^*((bc)^+)^*)^* = (a^*(bc)^*)^*$.

Соответствующее теоретико-множественное описание весьма громоздко:

$L = \{a^{m_1}(bc)^{n_1}a^{m_2}(bc)^{n_2}...a^{m_k}(bc)^{n_k} : (\forall i = \overline{1,k})(m_i, n_i \geq 0, k \geq 0\}$.

Позже мы увидим, что для некоторых регулярных языков их представление через коллективизирующие свойства практически невозможно.

В заключение этого параграфа рассмотрим пример тождественных преобразований регулярного выражения:

$(b^+a)^*(b^+a + b^*) =_{(7)} (b^+a)^*(b^+a) + (b^+a)^*b^* =_{(9)} (b^+a)^+ + (b^+a)^*b^* =_{(10)}$
$=_{(10)} (b^+a)^+ + (b^+a)^*(\lambda + b^+) =_{(7)} (b^+a)^+ + (b^+a)^* + (b^+a)^*b^+ =_{(14)} (b^+a)^* + (b^+a)^*b^+ =_{(7)}$
$=_{(7)} (b^+a)^*(\lambda + b^+) =_{(10)} (b^+a)^*b^*$.

(в скобках записаны номера используемых тождеств).

3. Конечные автоматы. Теорема Клини (анализ и синтез)

Определение КА как размеченного орграфа

Конечный автомат – это орграф, размеченный над полукольцом регулярных языков в некотором алфавите V, причем на функцию разметки $\varphi: E \to \mathrm{R}(V) \setminus \{0\}$ накладываются следующие ограничения: меткой дуги может быть либо некоторое подмножество алфавита V, либо язык, состоящий из одного пустого слова, причем, по определению, эти случаи разметки исключают друг друга.

На рисунках эту разметку дуг изображают так:

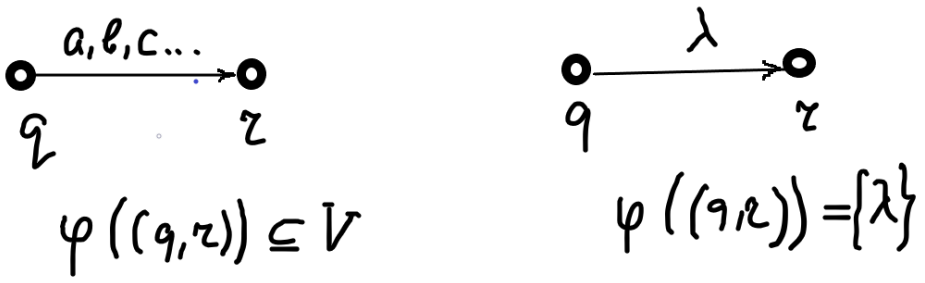


Рис. 1. Изображение дуг конечного автомата

Дугу с меткой λ будем дальше называть λ*-переходом*. Сразу принимается, что λ-переход не может быть петлей.

Заметим, что символ λ используется здесь как регулярное выражение, обозначающее язык, состоящий из одного пустого слова.

Кроме того, среди вершин графа выделяют одну вершину, называемую *начальной*, и некоторое подмножество вершин, называемых *заключительными*.

Вершины графа традиционно называют состояниями.

Таким образом, конечный автомат (далее будем использовать аббревиатуру КА) задается кортежем

$$M = (Q, E, \varphi, q_0, F),$$

где Q - множество состояний (вершин графа), E - множество дуг, φ - функция разметки, q_0 - начальное состояние, F - подмножество (возможно пустое) заключительных состояний.

Нам, однако, будет удобнее задавать КА такой упорядоченной пятеркой:

$$M = (V, Q, q_0, F, \delta),$$

где $\delta : Q \times (V \cup \{\lambda\}) \to 2^Q$ - отображение, называемое *функцией переходов*.

По определению $r \in \delta(q,a) \rightleftharpoons a \in \varphi((q,r)), a \in V \cup \{\lambda\}$.

Алфавит V называется входным алфавитом КА.

Функция переходов каждой упорядоченной паре (состояние, буква входного алфавита или пустое слово) сопоставляет некоторое множество состояний (может быть, и пустое) согласно записанному выше определению.

Функцию переходов удобно задавать в виде системы команд, где каждая команда имеет вид

$qa \to r$, где $r \in \delta(q,a)$.

При этом команда не записывается, если $\delta(q,a) = \varnothing$.

Пара слева от стрелки называется левой частью команды, а состояние справа от стрелки – правой частью команды.

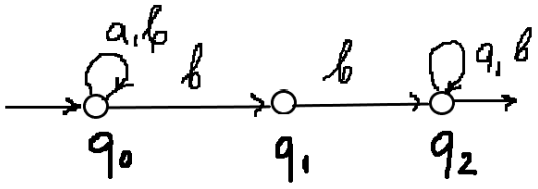


Рис. 2 КА $M_1 = (\{a,b\}, \{q_0, q_1, q_2\}, q_0, \{q_2\}, \delta_1)$

Система команд δ_1 автомата, изображенного выше на рисунке, записывается так:

$$\delta_1 : \begin{cases} q_0 a \to q_0 \\ q_0 b \to q_0 \mid q_1 \\ q_1 b \to q_2 \\ q_2 a \to q_2 \\ q_2 b \to q_2 \end{cases}$$

Разные команды с одинаковыми левыми частями принято записывать одной строкой, разделяя правые части вертикальной чертой.

Заметим, что в этом КА $\delta(q_1, a) = \varnothing$.

Начальное состояние отмечается входной стрелкой, а каждое заключительное – выходной. Эти стрелки – условные обозначения, и их ни в коем случае нельзя считать дугами графа

Перейдем теперь к определению языка, допускаемого КА.

Метка пути в КА как размеченном орграфе – соединение меток дуг, входящих в путь в порядке их прохождения (для пути ненулевой длины) и язык, состоящий из одного пустого слова (единица полукольца), обозначаемого буквой λ, для пути нулевой длины.

Поскольку метка каждой дуги – простой регулярный язык (либо конечное множество букв – однобуквенных слов, либо язык, состоящий из одного пустого слова), то метка любого пути конечной длины будет конечным и, стало быть, регулярным языком.

Например, меткой пути $q_0 \to q_0 \to q_1 \to q_2 \to q_2$ в КА на рис.... Будет язык, обозначаемый регулярным выражением $(a+b)bb(a+b) = abba + abbb + bbba + bbbb$.

Примем обозначение $q \Rightarrow *_x r \rightleftharpoons (\exists W : q \Rightarrow *r)(x \in \varphi^*(W))$ и будем говорить при этом, что слово x читается на некотором пути из состояния q в состояние r.

Также будем писать $q \to_a r, a \in V \cup \{\lambda\}$, если $r \in \delta(q,a)$, и говорить, что символ $a \in V \cup \{\lambda\}$

читается на дуге, ведущей из состояния q в состояние r (или, возможен переход из первого состояния во второе по соответствующему символу – букве входного алфавита или пустой цепочке).

Например, для КА на рис. 2: $q_0 \Rightarrow^*_{abba} q_2$, но $q_0 \not\Rightarrow^*_{aba} q_2$.

Стоимость прохождения из состояния q в состояние r:

$$c_{qr} = \bigcup_{W:q \Rightarrow *r} \varphi^*(W) = \{x : (\exists W : q \Rightarrow *r)(x \in \varphi^*(W))\},$$

но так как по определению $q \Rightarrow *_x r \rightleftharpoons (\exists W : q \Rightarrow *r)(x \in \varphi^*(W))$, то

$$c_{qr} = \{x : q \Rightarrow *_x r\}.$$

Тогда язык конечного автомата M есть, по определению

$$L(M) = \{x : q_0 \Rightarrow *_x q_f, q_f \in F\} = \bigcup_{q_f \in F} c_{q_0 q_f}.$$

Итак, ***язык КА (или язык, допускаемый КА) есть множество всех слов во входном алфавите, которые читаются на всех путях, ведущих из начального состояния в какое-либо из заключительных.***

Из определения языка КА следует, что для того, чтобы найти этот язык, достаточно найти матрицу стоимостей КА как размеченного орграфа. Но нам не нужна вся матрица стоимостей, а нужна только некоторая «вырезка» из нее, а именно те элементы, которые находятся в строке, номер которой равен номеру начального состояния, на пересечении с теми столбцами, номера которых соответствуют номерам заключительных состояний.

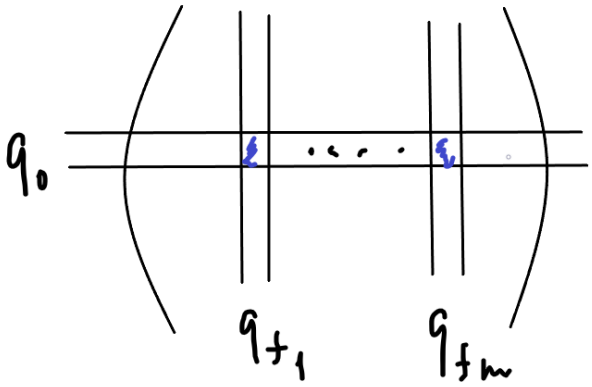


Рис. 3. Вырезка из матрицы стоимостей КА

Поэтому для нахождения языка КА достаточно решить только одну систему уравнений в полукольце регулярных языков.

А именно, можно показать (см. Учебник, п. 7.5, стр. 490), что система линейных уравнений для решения задачи о нахождении языка КА, называемой задачей анализа КА, имеет вид:

$$\xi = A\xi + \beta,$$

в которой компонента b_i столбца свободных членов определяется так:

$$b_i = \begin{cases} \lambda, q_i \in F \\ \varnothing, q_i \notin F \end{cases}$$

То есть в правых частях тех уравнениях системы, номера которых соответствуют номерам заключительных состояний, добавляется слагаемое λ (единица полукольца), а в остальных уравнениях свободный член равен $\varnothing$ (пустому языку – нулю полукольца). Язык обозначается той компонентой вектора решения, номер которой равен номеру начального состояния.

Для КА, изображенного на рис.2, система уравнений для нахождения языка имеет вид:

$$\begin{cases} x_0 = (a+b)x_0 + bx_1 \\ x_1 = bx_2 \\ x_2 = (a+b)x_2 + \lambda \end{cases}$$

Решить эту систему легко: достаточно заметить, что последнее уравнение содержит только неизвестное x_2, которое сразу определяется:

$$x_2 = (a+b)^* \lambda = (a+b)^*.$$

Из второго уравнения получаем

$$x_1 = bx_2 = b(a+b)^*.$$

Подставляя последнее выражение в первое уравнение, получим:

$$x_0 = (a+b)x_0 + bb(a+b)^*,$$

откуда

$$x_0 = (a+b)^* bb(a+b)^* = L(M_1)$$

Это и есть регулярное выражение, обозначающее язык КА M_1. Оно наглядно говорит о том, что этот язык есть множество всех таких слов а алфавите $\{a,b\}$, которые содержат вхождение двух букв “b” подряд.

Более интересный пример содержится в файле семинара №7 (в начале).

Лекция №17

20.11.24

3. Конечные автоматы. Теорема Клини (анализ и синтез) (продолжение)

Имеет место следующая важная теорема:

Теорема 1. Язык любого КА регулярен.

Доказательство. Язык КА есть компонента вектора решения линейной системы с регулярными коэффициентами[3].

Для доказательства используем индукцию по порядку n системы.

При $n=1$ система становится уравнением вида

$$x = ax + b,$$

где a и b - регулярные выражения.

Решение этого уравнения имеет вид $x = a^* b$, а так как итерация любого регулярного языка регулярна, то решение также будет регулярно.

Базис доказан.

Формулируя стандартно индукционное предположение, рассмотрим систему порядка n. Выражая неизвестное x_1 через остальные согласно формуле

$$x_1 = a^*_{11}(a_{12}x_2 + \ldots + a_{1n}x_n + b_1)$$

получим систему

$$x_k = (a_{k1}a^*_{11}a_{12} + a_{k2})x_2 + \ldots + (a_{k1}a^*_{11}a_{1k} + a_{kk})x_k + \ldots + \\ + (a_{k1}a^*_{11}a_{1n} + a_{kn})x_n + (a_{k1}a^*_{11}b_1 + b_k)$$

[3] Напомним, что везде в полукольцах (замкнутых и с итерацией) рассматриваются только наименьшие решения систем и уравнений.

при $k=2,...,n$, порядок которой равен $n-1$, а все ее коэффициенты регулярны. В силу индукционного предположения вектор решения этой системы состоит из регулярных языков. Согласно выражению для x_1 тогда и компонента X_1 решения будет регулярна.

Теорема доказана[4].

Замечание. Покажем более подробно подстановку выражения для X_1:

$$x_k = a_{k1}\underbrace{a^*_{11}(a_{12}x_2+...+a_{1k}x_k+...+a_{1n}x_n+b_1)}_{x_1}+a_{k2}x_2+...+a_{kk}x_k+...$$

$$...+a_{kn}x_n+b_k=(a_{k1}a^*_{11}a_{12}+a_{k2})x_2+...+(a_{k1}a^*_{11}a_{1k}+a_{kk})x_k+...+$$
$$+(a_{k1}a^*_{11}a_{1n}+a_{kn})x_n+a_{21}a^*_{11}b_1+b_k, 2\le k\le n.$$

Задача вычисления языка КА называется, как мы уже говорили, задачей анализа КА.

Ниже рассмотрены еще два примера решения этой задачи.

1. Найти язык, допускаемый следующим конечным автоматом:

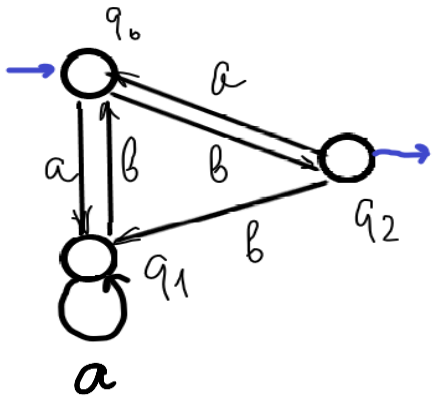


Рис. 4. Пример анализа КА

Система уравнений:

[4] Этот результат является частным случаем теоремы, согласно которой решение линейной системы с коэффициентами из любого полукольца с итерацией является вектором, все компоненты которого принадлежат этому же полукольцу. Другими словами, любое полукольцо с итерацией замкнуто относительно решений линейных систем с коэффициентами из этого полукольца. См. статью А.И. Белоусова «О некоторых свойствах полуколец», выложенную в облаке.

$$\begin{cases} x_0 = ax_1 + bx_2 \\ x_1 = bx_0 + ax_1 \\ x_2 = ax_0 + bx_1 + \lambda \end{cases}$$

Разумно на первой итерации исключить x_2 , поскольку это неизвестное уже выражено через остальные. Получаем:

$$\begin{cases} x_0 = ax_1 + b(ax_0 + bx_1 + \lambda) \\ x_1 = bx_0 + ax_1 \end{cases}$$

Приведем подобные члены в правой части первого уравнения:

$$\begin{cases} x_0 = bax_0 + (a + b^2)x_1 + b \\ x_1 = bx_0 + ax_1 \end{cases}$$

Из 2-го уравнения выражаем x_1:

$$x_1 = a * bx_0 .$$

Подставляя это выражение в первое уравнение, получим:

$$\{x_0 = bax_0 + (a + b^2)a * bx_0 + b .$$

Окончательно:

$$x_0 = (ba + a^+b + b^2a * b)x_0 + b ,$$

Откуда получаем регулярное выражение для языка:

$$L = x_0 = (ba + a^+b + b^2a * b) * b$$

Выражение в скобках описывает метки всех возможных замкнутых путей, проходящих через начальное состояние.
Рекомендуется сравнить это решение с приведенным в Учебнике на стр. 491 (по 7-му изд.).

2. Найти язык, допускаемый следующим КА:

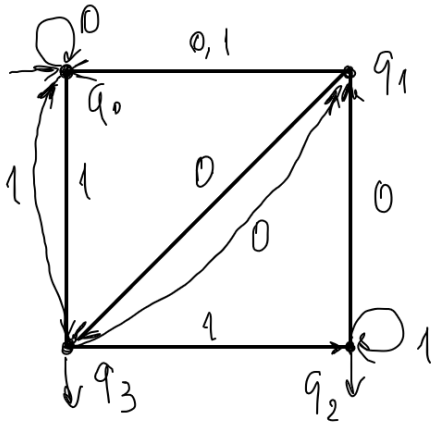


Рис. 5. Пример анализа КА.

Система уравнений:

$$\begin{cases} x_0 = 0x_0 + 1x_3 \\ x_1 = (0+1)x_0 + 0x_3 \\ x_2 = 0x_1 + 1x_2 + \lambda \\ x_3 = 1x_0 + 0x_1 + 1x_2 + \lambda \end{cases}$$

Ноль и единица понимаются, конечно, как символы, цифры, но никак не числа. Мы видим, что в системе x_1 и x_3 выражены через остальные неизвестные. Проще исключить x_1.
Перепишем систему:

$$\begin{cases} x_0 = 0x_0 + 1x_3 \\ x_2 = 0((0+1)x_0 + 0x_3) + 1x_2 + \lambda \\ x_3 = 1x_0 + 0((0+1)x_0 + 0x_3) + 1x_2 + \lambda \end{cases}$$

Приводим подобные:

$$\begin{cases} x_0 = 0x_0 + 1x_3 \\ x_2 = 0(0+1)x_0 + 1x_2 + 00x_3 + \lambda \\ x_3 = (1 + 0(0+1))x_0 + 1x_2 + 00x_3 + \lambda \end{cases}$$

Теперь исключаем x_2:

$$x_2 = 1^*\left(0(0+1)x_0 + 00x_3 + \lambda\right)$$

Преобразуем теперь уравнение для x_3:

$$x_3 = (1 + 0(0+1))x_0 + 1 \cdot 1^* \left(0(0+1)x_0 + 00x_3 + \lambda\right) + 00x_3 + \lambda;$$

$$x_3 = (1 + 0(0+1) + 1^+0(0+1))x_0 + (1^+00 + 00)x_3 + 1^+ + \lambda.$$

Вынося в коэффициентах при неизвестных общие множители (справа!) и учитывая, что для любого a итерация $a^* = \lambda + a^+$, получим:

$$x_3 = (1 + (\lambda + 1^+)0(0+1))x_0 + (1^+ + \lambda)00x_3 + 1^*,$$

$$x_3 = (1 + 1^*0(0+1))x_0 + 1^*00x_3 + 1^*$$

Полезно заметить, что выражения 1*0(0+1) и 1*00 дают метки соответствующих путей как через вершину q2, так и минуя ее (когда итерация замещается пустой цепочкой). Точно также свободный член 1* показывает, что из вершины q3 можно выйти сразу или пройти по 1 в q2 и, покрутившись там по петле сколько угодно раз, или ни разу, выйти из нее.

Выражаем x_3 через x_0:

$$x_3 = (1^*00)^*((1 + 1^*0(0+1))x_0 + 1^*)$$

В итоге получаем уравнение для x_0:

$$x_0 = 0x_0 + 1(1^*00)^*((1 + 1^*0(0+1))x_0 + 1^*),$$

$$x_0 = (0 + 1(1^*00)^*(1 + 1^*0(0+1)))x_0 + 1(1^*00)^*1^{*\prime}$$

откуда получаем выражение для языка:

$$L = x_0 = (0 + 1(1^*00)^*(1 + 1^*0(0+1)))^*1(1^*00)^*1^*$$

Выражение в скобках, подвергаемое итерации, описывает все возможные «кручения» в начальном состоянии. Можно крутиться по петле (метка 0), или пройти по 1 в q3, пройти там по контурам q3→q1→q3 или q3→q2→q1→q3, и вернуться в начало либо сразу по 1, либо опять-таки через q1 или q2 и q1. Выражение 1(1*00)*1* описывает все возможности выхода: либо сразу по 1 перейти в q3 и выйти, либо, снова покрутившись там по указанным контурам, выйти, либо выйти из q2, перейдя в эту вершину по 1.

Имеет место теорема, обратная к теореме 1:

Теорема 2. Для каждого регулярного языка может быть построен КА, который его допускает.

Доказательство. Опишем построение новых конечных автоматов из уже построенных, следуя индуктивному определению множества регулярных языков.

Ниже на рисунках представлены автоматы для элементарных регулярных языков и конструкции для объединения, соединения, итерации и позитивной итерации.

Доказательство корректности этих построений мы не рассматриваем. Впрочем, они достаточно прозрачны.

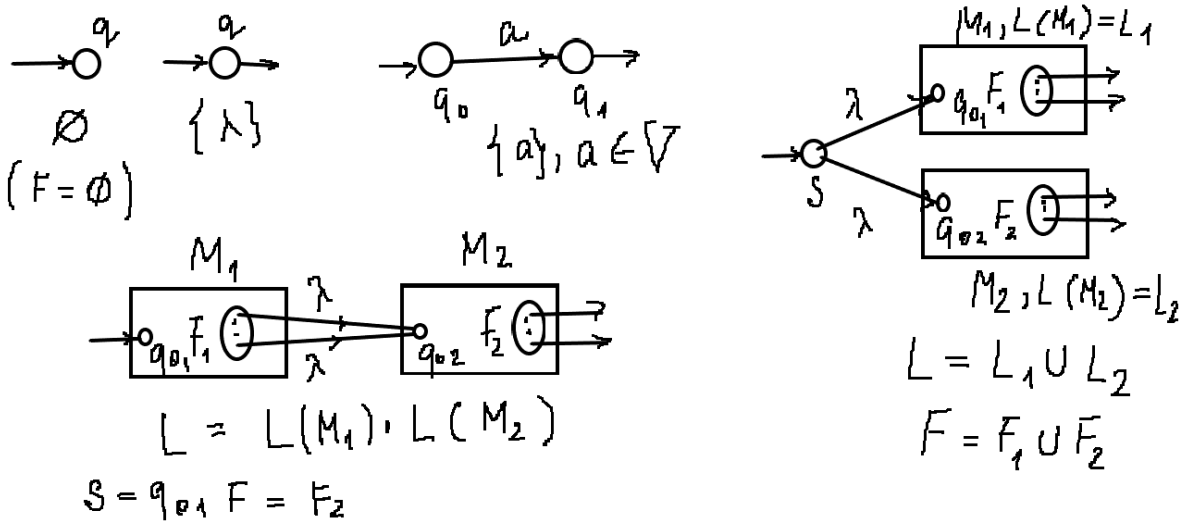
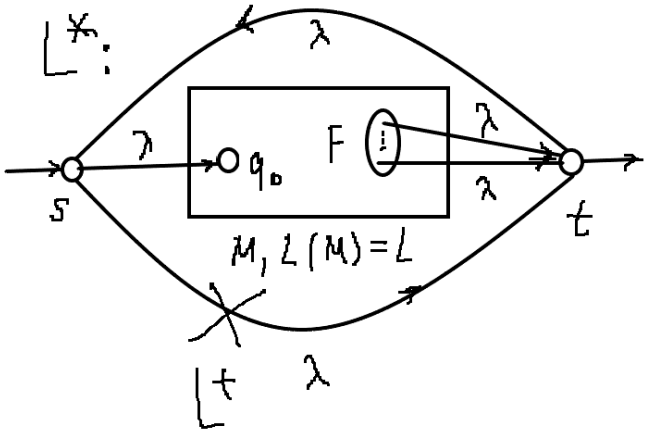


Рис. 6. КА для элементарных регулярных языков, объединения и соединения.



Если удалить дугу (s, t), получится КА для позитивной итерации языка L.

Рис. 7. КА для итерации и позитивной итерации языка.

Следует обратить внимание на то, что в КА для итерации в общем случае необходимо вводить новые начальную и заключительную вершины. Это абсолютно необходимо, если в «итерируемом» КА через его начальную или конечную вершину проходит контур. Если это правило не соблюдать, можно получить ошибочное решение.

(См. в Учебнике замечание 7.19, стр. 488).

Очень подробно особенности при построении конструкций из автоматов рассмотрены в файле семинара №7 при разборе задачи по образцу ДЗ.

Объединяя теоремы 1 и 2, получаем утверждение, называемое теоремой Клини:

Теорема Клини. Язык регулярен тогда и только тогда, когда он допускается некоторым конечным автоматом.

Эта теорема дает исчерпывающую характеристику класса регулярных языков.

Задача же построения КА по данному регулярному выражению называется задачей синтеза КА.

Ниже приведен пример решения такой задачи.

$$L = (a^* + (bc)^+)^*,$$
$$L = (\{a^n : n \geq 0\} \cup \{(bc)^n : n \geq 1\})^*,$$
$$L = \{a^{m_1}(bc)^{n_1} a^{m_2}(bc)^{n_2} \ldots a^{m_k}(bc)^{n_k} : (\forall i = \overline{1,k})(m_i, n_i \geq 0, k \geq 0\}.$$

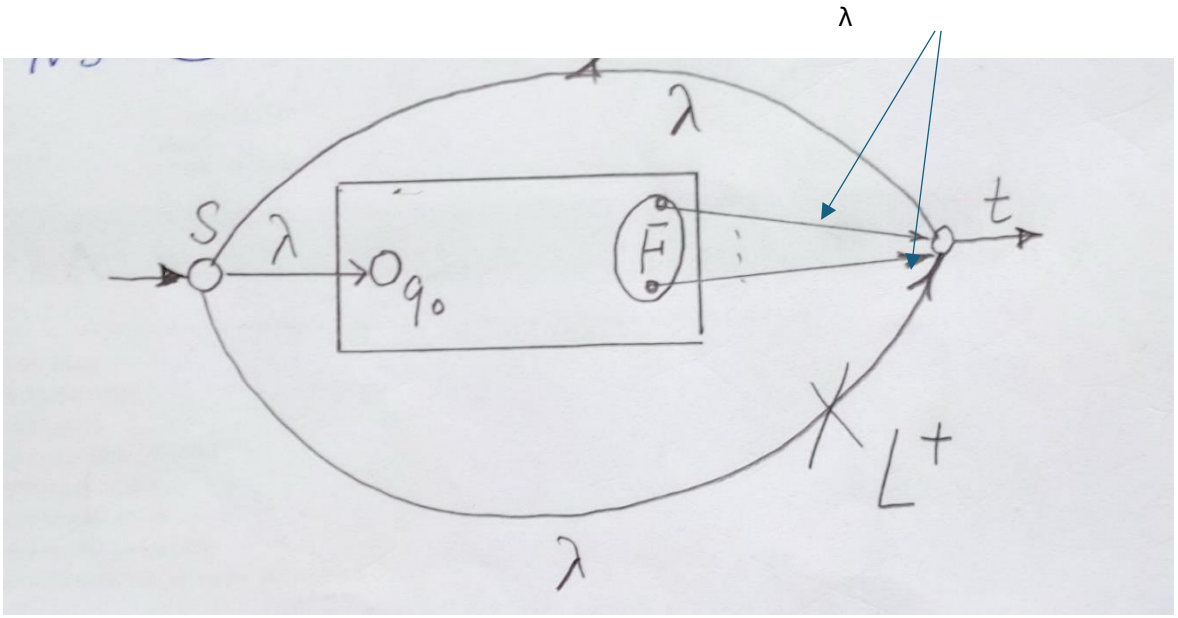


Рис. 8. КА для итерации и позитивной итерации

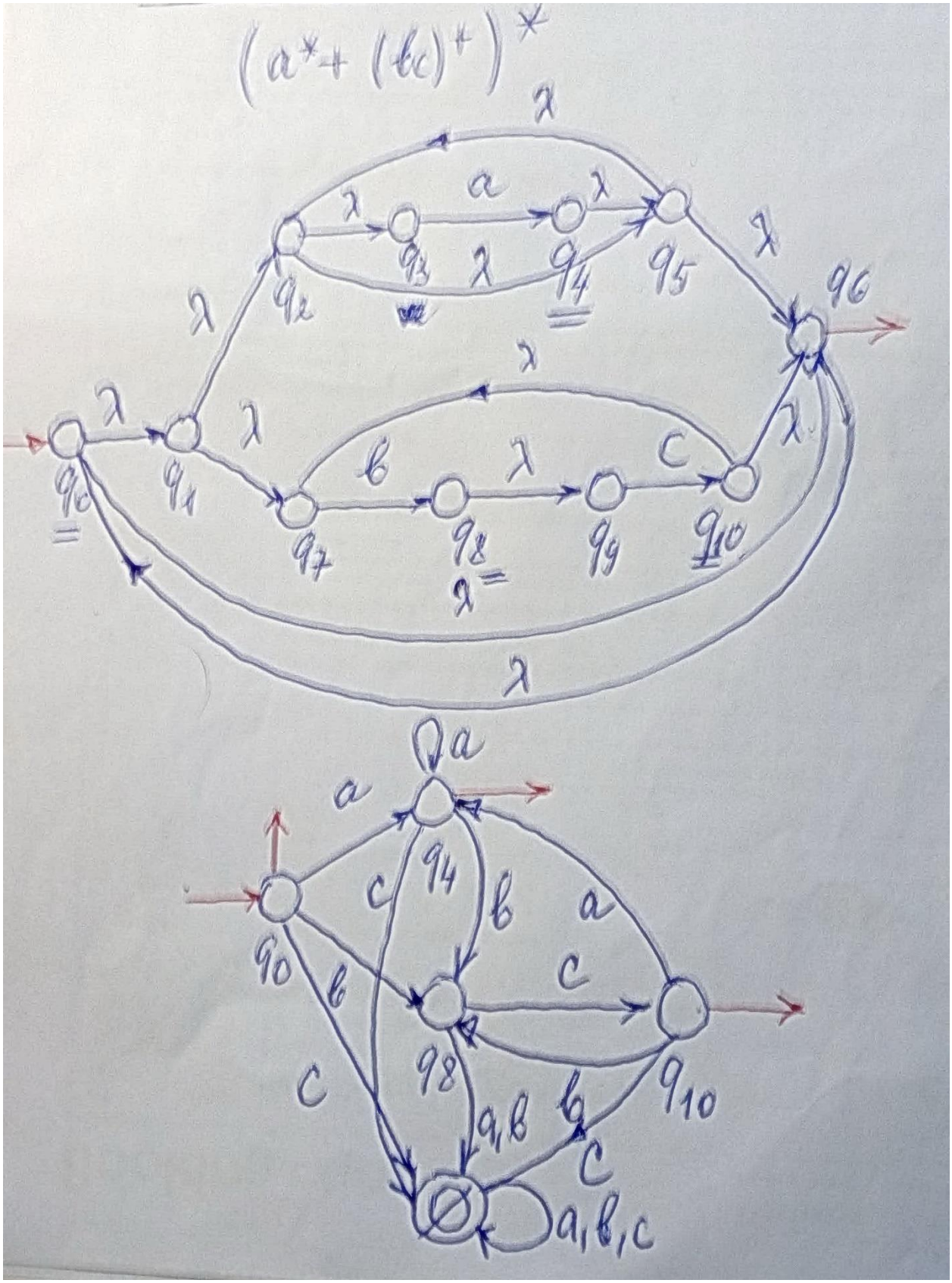


Рис. 9. Пример синтеза.

Используя схему синтеза КА, можно доказать регулярность любого конечного языка, построив для него допускающий КА (решить эту задачу самостоятельно!).

3. Детерминизация КА. Регулярность дополнения регулярного языка

Автомат, построенный по алгоритму синтеза, требует некоторой оптимизации. Эта оптимизация предполагает сначала т.н. детерминизацию с предварительным удалением λ-переходов и дальнейшую минимизацию по числу состояний. Последняя в нашем курсе не рассматривается (см. Учебник, п. 7.7).

Перед тем, как рассматривать алгоритм детерминизации, необходимо ввести некоторые определения.

Эквивалентные КА

Два КА называются эквивалентными, если они допускают один и тот же язык:

$$M_1 \simeq M_2 \rightleftharpoons L(M_1) = L(M_2)$$

Детерминированный КА

КА называется *детерминированным*, если: 1) в нем нет λ-переходов и 2) для любых состояния q и буквы a входного алфавита множество $\delta(q,a)$ содержит ровно один элемент (проще говоря, из каждого состояния по каждой букве входного алфавита определен переход ровно в одно состояние).

Если же, при отсутствии λ-переходов, для любых состояния q и буквы a входного алфавита множество $\delta(q,a)$ содержит не более одного состояния (то есть это множество либо пусто, либо содержит ровно один элемент), то такой КА называется *квазидетерминированным*.

В квазидетерминированном КА не может быть разных дуг, исходящих из какого-либо состояния и помеченных одинаковыми буквами, но могут быть переходы, которые не определены (то есть соответствующее значение функции переходов есть пустое множество).

В детерминированном КА функцию переходов можно задать как отображение

$$\delta : Q \times V \to Q.$$

Теорема (о детерминизации КА). Для любого КА может быть построен эквивалентный ему детерминированный КА (ДКА).

Доказательство проведем, описав алгоритм построения эквивалентного ДКА. Доказательство корректности алгоритма (не очень простое) не рассматривается (см. Учебник, п. Д7.3).

Детерминизация проводится в два этапа. Сначала удаляются λ-переходы, после чего проводится собственно детерминизация.

Формулы удаления λ-переходов

По исходному КА:

$$M = (V, Q, q_0, F, \delta).$$

строится эквивалентный КА без λ-переходов:

$$M' = (V, Q', q'_0, F', \delta'),$$

где

$$Q' = \{q_0\} \cup \{r : (\exists q \in Q)(\exists a \in V)(r \in \delta(q,a))\},$$

то есть в новом автомате остаются вместе с начальным состоянием только те состояния, в которые заходит хотя бы одна дуга, помеченная буквами входного алфавита;

$q'_0 = q_0$ (начальное состояние остается прежним);

$$F' = (F \cap Q') \cup \{r : (\exists q \in F)(M : r \Rightarrow *_\lambda q)\};$$

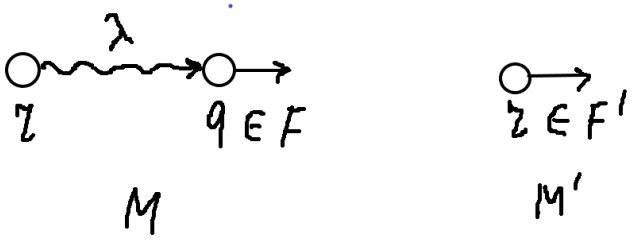


Рис. 10. Изменение множества заключительных состояний при удалении λ-переходов.

К множеству заключительных состояний исходного КА, которые остаются в новом КА, добавляются те, из которых в исходном КА можно пройти в заключительное состояние по λ-переходам.

$$\delta'(q,a) = \delta(q,a) \cup$$
$$\cup\{p : (\exists r \in Q)(M : q \Rightarrow^*_\lambda r) \wedge (p \in \delta(r,a))\}$$

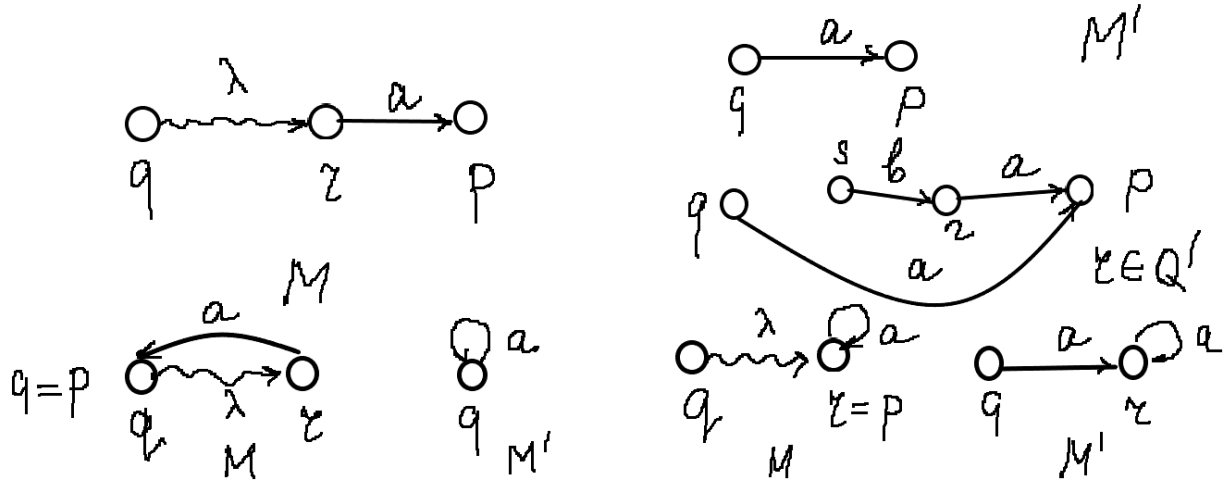


Рис. 11. Преобразование графа КА при удалении λ-переходов.

Это самое важное преобразование исходного графа. В нем надо найти все такие тройки состояний (q, r, p), что из первого во второе можно пройти по λ-переходам, а из второго в третье по некоторой букве $a \in V$ входного алфавита. Тогда в новом КА надо провести дугу с меткой $a \in V$ из первого состояния в третье. Это понятно: ведь в исходном КА можно по λ-переходам, то есть ничего не читая, пройти из первого состояния во второе, а уже из него в третье по букве $a \in V$. В итоге из первого в третье можно пройти, прочитав букву $a \in V$. Ясно, что после удаления λ-переходов надо в новом КА задать прямой переход из первого состояния в третье по букве $a \in V$.

При этом второе, промежуточное, состояние может остаться в новом КА, а может и исчезнуть. Последнее происходит, если в это состояние заходят только λ-переходы.

Возможно, что первое состояние совпадет с третьим ($q = p$), то есть из r в q ведет «обратная» дуга, замыкающая путь по λ-переходам в обратном направлении. Тогда в новом автомате возникает петля в состоянии q.

Могут совпасть второе и третье состояния $r = p$. Тогда петля, помеченная буквой $a \in V$ в состоянии $r = p$ остается и добавляется новая дуга с этой же буквой, ведущая из q в $r = p$. Первое и второе состояния совпасть не могут, так как петель с меткой λ не может быть по определению.

Лекция №18

22.11.24

Собственно детерминизация

По исходному КА без λ-переходов

$$M = (V, Q, q_0, F, \delta)$$

строится детерминированный КА

$$M' = (V, Q', q'_0, F', \delta'),$$

параметры которого определяются следующим образом:

$$Q' = 2^Q, q'_0 = \{q_0\}, F' = \{T : T \cap F \neq \varnothing\},$$
$$\delta'(S, a) = \bigcup_{q \in S} \delta(q, a).$$

Требуются некоторые пояснения.

Каждое состояние нового КА соответствует некоторому подмножеству множества состояний исходного КА, но не следует думать, что оно как-то «рассыпается» на отдельные элементы. Нет, это единое и неделимое состояние. Ведь множеством состояний любого КА может быть любое конечное множество, и если множество Q конечно (а мы только такие автоматы и рассматриваем), то и его булеан 2^Q конечен. Для краткости будем состояния КА M' *состояниями-множествами*.

Начальное состояние нового КА есть одноэлементное множество, состоящее из начального состояния исходного КА.

Заключительными состояниями нового КА будут те и только те состояния-множества, которые содержат хотя бы одно заключительно состояние исходного КА.

Новая функция переходов определяет переход из состояния-множества S в такое состояние-множество, которое равно объединению значений функции переходов исходного КА для всех состояний множества S (по заданной букве a). Поскольку это состояние-множество единственное, то новый КА построен как детерминированный.

Построение детерминированное КА «методом вытягивания» подробно описано в файле семинара №7.

Ниже на рисунке изображен детерминированный КА, эквивалентный КА на рис. 2:

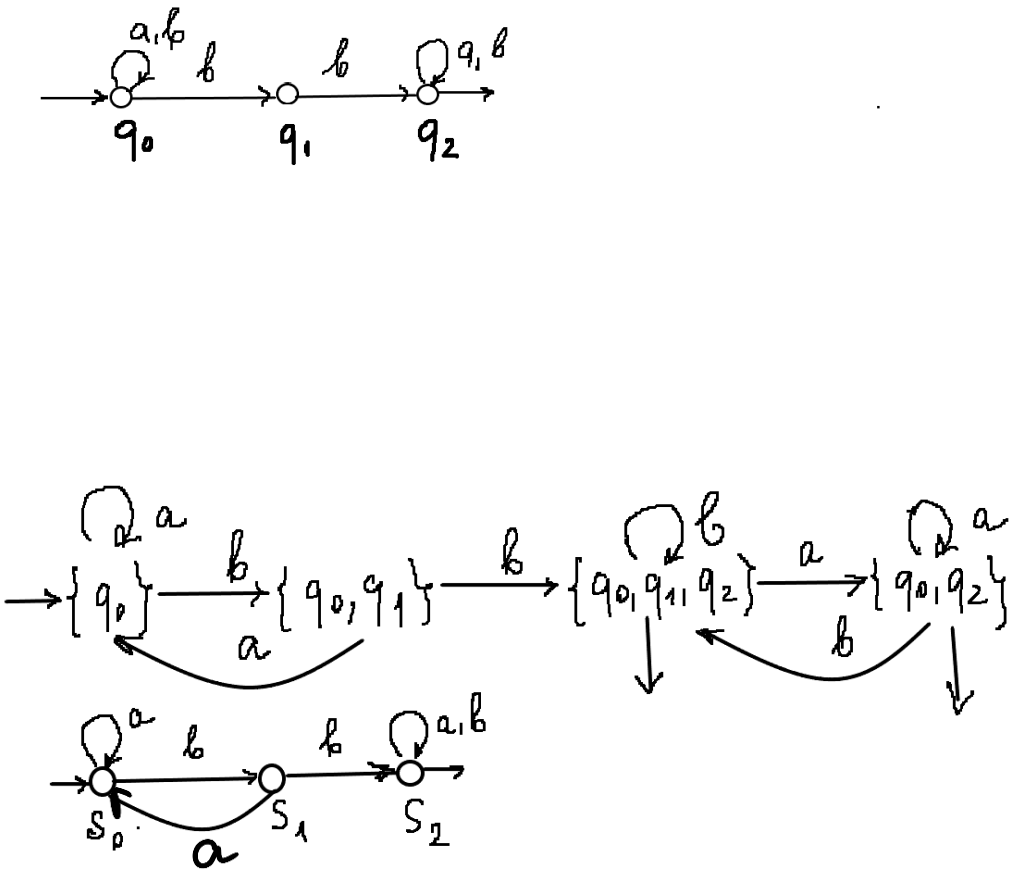


Рис. 12. Пример детерминизации КА.

Значение функции переходов нового КА вычисляются последовательно до «насыщения», то есть до тех пор, пока появляются новые состояния-множества.

$$\delta'(\{q_0\},a)=\{q_0\},\delta'(\{q_0\},b)=\{q_0,q_1\};$$

$$\delta'(\{q_0,q_1\},a)=\delta(q_0,a)\cup\delta(q_1,a)=\{q_0\}\cup\varnothing=\{q_0\},$$

$$\delta'(\{q_0,q_1\},b)=\delta(q_0,b)\cup\delta(q_1,b)=\{q_0,q_1\}\cup\{q_2\}=$$

$$=\{q_0,q_1,q_2\};$$

$$\delta'(\{q_0,q_1,q_2\},a)=\delta(q_0,a)\cup\delta(q_1,a)\cup\delta(q_2,a)=$$

$$=\{q_0,q_2\};$$

$$\delta'(\{q_0,q_1,q_2\},b)=\delta(q_0,b)\cup\delta(q_1,b)\cup\delta(q_2,b)=$$

$$=\{q_0,q_1\}\cup\{q_2\}\cup\{q_2\}=\{q_0,q_1,q_2\};$$

$$\delta'(\{q_0,q_2\},a)=\{q_0,q_2\},\delta'(\{q_0,q_2\},b)=\{q_0,q_1,q_2\}.$$

Тем самым «вытянуты» все состояния-множества, достижимые из начального состояния. Остальные состояния нас не интересуют.

Можно заметить, что два заключительных состояния в детеминированном КА образуют т.н. поглощающее множество состояний: попав хотя бы в одно из них, автомат это множество никогда не покинет. Значит, эти два состояния можно «склеить» в одно заключительное, зациклив его по обеим буквам входного алфавита.

Переобозначив состояния более удобно для чтения, получим КА, который оказывается минимальным (по числу состояний) детерминированным КА, эквивалентным исходному.

Теорема о детерминизации имеет важное теоретическое следствие.

Теорема (о регулярности дополнения регулярного языка). Дополнение регулярного языка есть регулярный язык.

Доказательство. Пусть $L \subseteq V^*$ - регулярный язык в алфавите V. Докажем, что его дополнение $\overline{L} = V^* \setminus L$ - тоже регулярный язык.

Так как L регулярный язык, может быть построен КА $M = (V, Q, q_0, F, \delta)$, допускающий этот язык, причем, по тереме о детерминизации его можно считать детерминированным.

Возьмем произвольное слово $x = x(1)...x(k), k \geq 0$ в алфавите V. Так как КА M детерминированный, существует единственный путь в нем, начинающийся в начальном состоянии, на котором читается слово x:

$$q_0 \xrightarrow{x(1)} p_1 \xrightarrow{x(2)} p_2 ... \xrightarrow{x(k)} p_k$$

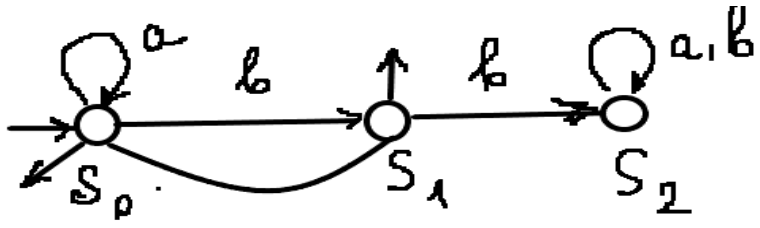
(при $k = 0$ $x = \lambda$ и принимается, что $p_0 = q_0$).

Ясно, что $x \in L \Leftrightarrow p_k \in F$.

Следовательно, для того, чтобы построить КА, допускающий в точности те слова, которые не допускает КА M, достаточно в нем поменять ролями заключительные и незаключительные состояния, то есть все заключительные состояния КА M сделать незаключительными, а незаключительные – заключительными. Таким образом, КА для дополнения языка L будет иметь вид:

$$\overline{M} = (V, Q, q_0, \overline{F} = Q \setminus F, \delta), L(\overline{M}) = \overline{L}.$$

Ниже показан КА, допускающий дополнение языка, допускаемого КА, показанного на рис. 2 и 12.



Заодно можно проанализировать этот автомат, то есть найти его язык.

Система уравнений:

$$\begin{cases} x_0 = ax_0 + bx_1 + \lambda \\ x_1 = ax_0 + bx_2 + \lambda \\ x_2 = (a+b)x_2 \end{cases}$$

Из последнего уравнения сразу получаем $x_2 = \varnothing$.

Подставляя это в систему, получим:

$$\begin{cases} x_0 = ax_0 + bx_1 + \lambda \\ x_1 = ax_0 + \lambda \end{cases}$$

Дальше:

$$x_0 = ax_0 + b(ax_0 + \lambda) + \lambda,$$
$$x_0 = (a + ba)x_0 + b + \lambda,$$

откуда получаем регулярное выражение, обозначающее язык в алфавите $\{a, b\}$, слова которого не содержат двух букв b подряд:

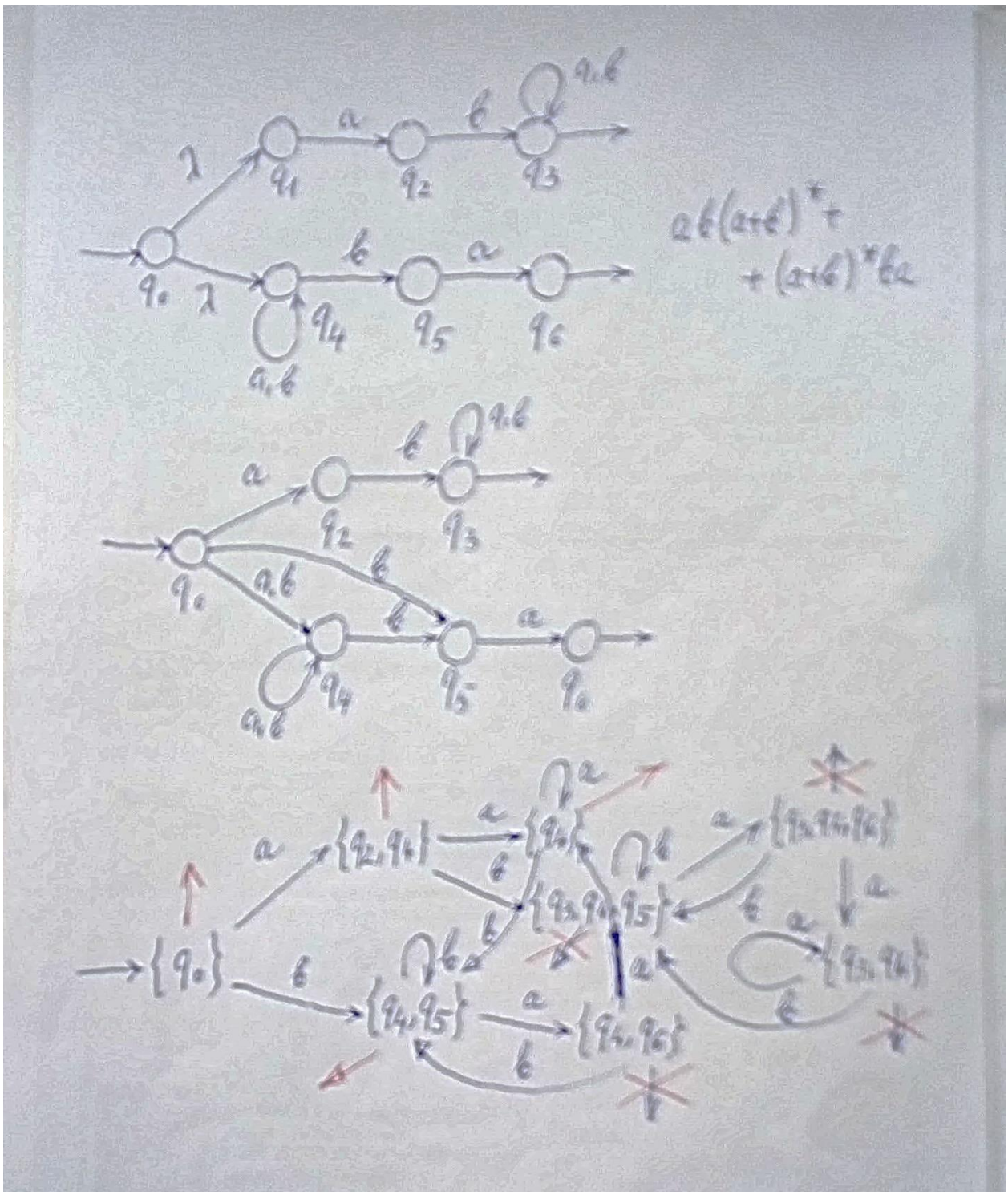
$$x_0 = (a + ba)^*(b + \lambda).$$

Приведем еще решение одной задачи, связанное с детерминизацией и последующим переходом к дополнению.

Пример.

Построить КА, который допускает те и только те слова в алфавите {a, b}, которые не начинаются на ab и не кончаются на ba.

$$L = \overline{ab(a+b)^*} \cap \overline{(a+b)^*ba} = \overline{ab(a+b)^* + (a+b)^*ba}$$



Предлагается самостоятельно решить такую задачу:

Построить КА, который допускает все цепочки в алфавите {a, b}, кроме тех, которые **одновременно** начинаются на ab и кончаются на ba.

Указание: строим КА для дополнения языка L=ab(a+b)*ba+aba.

Алгебраические свойства регулярных языков. Разрешимые проблемы.

Из теоремы о регулярности дополнения регулярного языка вытекает важное следствие:

Следствие. Вместе с любыми двумя регулярными языками регулярны будут их пересечение, разность и симметрическая разность.

Доказательство вытекает из известных теоретико-множественных тождеств:

$$L_1 \cap L_2 = \overline{\bar{L}_1 \cup \bar{L}_2}; L_1 \setminus L_2 = L_1 \cap \bar{L}_2; L_1 \Delta L_2 = (L_1 \setminus L_2) \cup (L_2 \setminus L_1).$$

Таким образом, множество регулярных языков замкнуто относительно всех рассмотренных ранее операций над множествами.

На основании этих алгебраических свойств регулярных языков можно решить проблему распознавания эквивалентности двух произвольных КА.

Но предварительно надо решить для КА проблему пустоты: является ли пустым язык, допускаемый данным КА?

Эта проблема решается простым поиском в ширину (алгоритмом волнового фронта) из начального состояния: язык КА пуст тогда и только тогда, когда в нем из начального состояния не достижимо ни одно заключительное. Ясно, что язык КА пуст и тогда, когда множество заключительных состояний пусто.

Тогда, в силу известного критерия равенства множеств $L_1 = L_2 \Leftrightarrow L_1 \Delta L_2 = \varnothing$ два КА эквивалентны тогда и только тогда, когда язык КА для симметрической разности их языков пуст.

В вычислительном отношении алгоритм построения КА для симметрической разности языков двух заданных КА может оказаться весьма сложным, но нам сейчас важна принципиальная разрешимость такой задачи.

4. Лемма о разрастании для регулярных языков

Чтобы доказать, что язык регулярен, достаточно как-то построить КА, который его допускает (или, по крайней мере, указать способ такого построения). При этом, конечно, надо доказать, что построенный КА допускает именно тот язык, который задан, то есть допускает все слова языка и только их[5].
Но как доказать, что язык не регулярен?
Существует важное необходимое условие регулярности языка, называемое леммой о разрастании (или леммой о накачке). Это условие конструктивно проверяемо и тем особенно важно.

Теорема 1 (*Лемма о разрастании для регулярных языков***).** Для любого регулярного языка L определена константа k_L (зависящая от L) такая, что всякая цепочка x языка L, длина которой не меньше k_L, представима в виде соединения трех цепочек: $x = uvw$, где цепочка v не пуста, ее длина не превосходит k_L, и для любого неотрицательного целого n цепочка $x_n = uv^n w$ принадлежит языку L.

Доказательство. Так как язык L регулярен, можно построить допускающий его КА

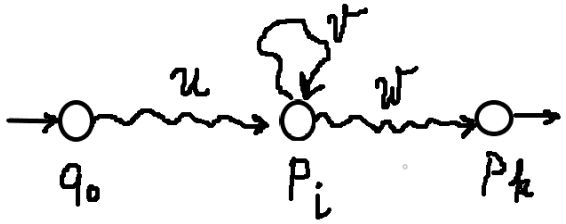
[5] В теории формальных языков доказывается важный критерий регулярности (необходимое и достаточное условие), известное как теорема Майхилла-Нероуда, опираясь на которую, можно доказать регулярность языка, не прибегая к построению КА. См. в Облаке статьи Белоусова А.И. с соавторами по этой теме.

$$M = (V, Q, q_0, F, \delta),$$

который можно считать детерминированным. Введем константу $k_L = |Q|$, то есть выберем эту константу как число состояний указанного автомата. Возьмем слово (цепочку) $x = x(1)...x(k), k \geq |Q|$. Это слово читается на единственном пути, ведущем из начального состояния в заключительное:

$$q_0 \xrightarrow{x(1)} p_1 \xrightarrow{x(2)} p_2 ... \xrightarrow{x(k)} p_k \in F$$

Длина пути равна длине слова x, а поскольку число вершин в любом пути в графе на единицу больше числа дуг, то число состояний в этом пути будет больше числа всех состояний. Отсюда следует, что по крайней мере какое-то одно состояние будет повторяться в пути. Повторяющееся состояние лежит на некотором замкнутом пути и, следовательно, на простом замкнутом пути, то есть на контуре. Значит, представленный выше путь разделяется на три части:



Тогда и слово x разделяется на три части:

$$x = uvw,$$

где цепочка u читается на пути из начального состояния до повторяющейся вершины $p_i, 0 \leq i \leq k, p_0 = q_0$, цепочка v читается на контуре, а цепочка w - на пути из повторяющейся вершины до заключительной.

Но контур можно обходить сколько угодно раз, а можно вообще его пропустить, откуда следует, что любое слово $x_n = uv^n w, n \geq 0$ читается на некотором пути из начального состояния в заключительное, то есть $(\forall n \geq 0)(x_n = uv^n w \in L)$.

Таким образом, возможность «накачки» (разрастания – повторения любого числа раз; или даже выбрасывания) средней части слова x доказана.

Слово v, читаемое на контуре, не может быть пустым, так как контур – путь ненулевой длины. Поскольку контур – *простой* замкнутый путь и только одна вершина в нем может встретиться дважды, наибольшая длина его равна числу вершин графа, откуда и вытекают ограничения на длину «накачиваемой» цепочки: $0 < |v| \leq k_L = |Q|$.

Теорема доказана.

Переходим теперь к разбору примеров на применение леммы о разрастании к анализу конкретных языков.

Общая схема применения леммы о разрастании для доказательства нерегулярности конкретного языка такова.

Доказательство от противного: предполагаем, что язык регулярен и, согласно лемме, представляем каждое его достаточно длинное слово в виде соединения трех слов $x = uvw$, после чего анализируем возможные случаи расположения средней («накачиваемой») цепочки. Каждый такой случай должен быть отвергнут как приводящий к противоречию с условием леммы. Если хотя бы один такой случай расположения средней цепочки не удалось доказательно отвергнуть, то задача не решена.

Некоторые языки бывает трудно, а то невозможно, проанализировать с помощью самой леммы, и тогда приходится прибегать к использованию других средств. Например, можно исходный язык пересечь с некоторым регулярным языком и уже с помощью леммы доказать нерегулярность полученного пересечения. Тогда можно сделать вывод и о нерегулярности исходного языка, опираясь на теорему о регулярности пересечения регулярных языков.

Заметим попутно, что анализ регулярности/нерегулярности языка имеет смысл только для бесконечных языков, так как любой конечный язык регулярен и для такого языка лемма о разрастании выполняется тривиально, так как можно показать, что слов с указанным в лемме ограничением на длину в конечном языке просто не существует.

Переходим к рассмотрению примеров.

1) $L_1 = \{a^n b^n : n \geq 0\}$

Пусть этот язык регулярен. Тогда по лемме о разрастании существует некоторая константа k_L такая, что любое слово языка, длина которого не меньше этой константы, можно представить в виде соединения указанных в условии леммы трех слов:

$$x = a^n b^n = uvw$$

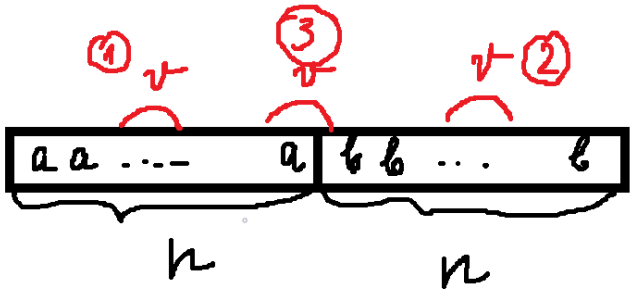
Можно считать, что число n выбрано так, что $n \geq k_L$.

Здесь есть один тонкий момент, который надо постараться понять. Существование константы k_L вытекает из предположения о регулярности рассматриваемого языка. Мы не можем знать, чему она равна, но можем быть уверены в том, что где-то на числовой прямой она фиксирована. Значит, ее можно превзойти, и параметры цепочки языка, определяющие ее длину, можно непротиворечиво полагать не меньше этой константы, так как длина слов языка не ограничена сверху.

Возвращаемся к анализу языка L_1.

Цепочку v можно расположить в слове языка тремя способами: 1) в «зоне» символов a, то есть $v = a^k, 0 < k \leq n$, 2) в зоне символов b, то есть $v = b^k, 0 < k \leq n$ и 3) на «стыке» двух зон: $v = a^k b^l, 0 < k, l \leq n$.

Это можно наглядно изобразить так:



$$L = \{a^n b^n : n \geq 0\}$$

Тогда в первом случае повторение («накачка») слова v приведет к нарушению равенства чисел вхождений букв в слово. Точно по такой же причине невозможен второй способ.

В третьем же имеем $v^2 = a^k b^l a^k b^l$, то уже первое повторение накачиваемой цепочки приводит к появлению вхождения слова ba, которого не может быть в словах языка L_1.

Таким образом, этот язык не удовлетворяет лемме о разрастании и потому нерегулярен.

2) $L_2 = \{x : x \in \{a,b\}^*, n_a(x) = n_b(x)\}$,

где $n_\alpha(x), \alpha \in \{a,b\}$ означает число вхождений буквы α в слово x.

То есть этот язык состоит из всех таких в алфавите $\{a,b\}$, что числа вхождений каждой буквы одинаковы, но буквы могут как угодно перемешиваться.

Рассмотрим тогда пересечение $L_2 \cap a^*b^* = L_1$.

Но мы только что доказали, что язык L_1 нерегулярен. Значит, нерегулярен и язык L_2, так как если бы он был регулярен, его пересечение с регулярным было бы регулярно, но оно не регулярно.

К такому «приему пересечения» следует прибегать в том случае, когда в пересечении исходного языка с некоторым регулярным получается более простой по структуре язык, к которому уже легко применить лемму о разрастании.

Но может оказаться, что к исходному языку лемма вообще не применима, более того, он может лемме удовлетворять, не будучи регулярным.

3) $L_3 = \{a^n b^m a^n : m,n \geq 0\}$.

Здесь очевидно, что нельзя отвергнуть расположение накачиваемой цепочки в зоне символов b. Но язык нерегулярен, что доказывается рассмотрением пересечения

$$L_3 \cap a^*ba^* = \{a^n ba^n : n \geq 0\}.$$

Нерегулярность полученного пересечения доказывается легко: ясно, что в зонах символов a (как в левой, так и в правой) накачиваемую цепочку разместить нельзя, но ее

нельзя разместить и с «захватом» буквы b, то есть полагая $v = a^k b a^l; 0 \le k,l \le n$, так тогда при накачке число букв b станет больше единицы, что недопустимо в полученном пересечении.

Но оказывается, что исходный язык даже удовлетворяет лемме о разрастании.

Чтобы доказать это, необходимо указать выбор константы k_L, а затем проверить возможность накачки. Здесь можно положить:

$$u = a^n b^k, v = b, w = b^l a^n, k + l + 1 = m,$$

если цепочка содержит хотя бы одну букву b; для цепочки a^{2n} можно считать $v = aa$. Это значит, что можно принять $k_L = 2$. Т. е. если $|x| = 2$, то $x = aa$, а самая короткая цепочка, длины, не меньшей 2, с буквой b имеет вид aba.

Сам факт, что существует нерегулярный язык, удовлетворяющий лемме о разрастании, не должен удивлять, так как условие леммы необходимо для регулярности, но не достаточно.

4) $L_4 = \{x \in V^* : x = x^R, |V| > 1\}$, где x^R - инверсия слова x, то есть если $x = x(1)x(2)...x(k-1)x(k)$, то $x^R = x(k)x(k-1)...x(2)x(1)$, причем $\lambda^R = \lambda$ (по определению).

Это язык палиндромов, то есть слов, совпадающих со своей инверсией: шалаш, шабаш...

Известен такой латинский палиндром (и магический квадрат):

SATOR

AREPO

TENET

OPERA

ROTAS

(Перевод: «Пахарь Арепо тянет рабочие плуги».)

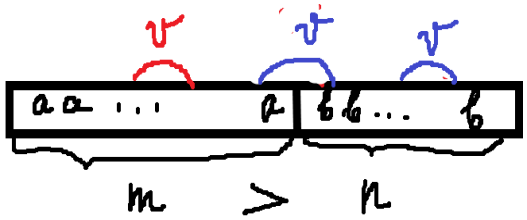
Нерегулярность языка палиндромов в алфавите, содержащем не менее двух букв, доказывается рассмотрением пересечения

$$L_4 \cap a^* b a^* = \{a^n b a^n : n \ge 0, a \ne b\}.$$

(см. предыдущий пример).

5) $L_5 = \{a^m b^n : m > n \ge 0\}$

Можно прибегнуть к той же квазигеометрической иллюстрации, что и в первом примере:



$$L = \{a^m b^n : m > n \geq 0\}$$

Но здесь зона символов a длиннее зоны символов b. Легко понять, что накачка в правой зоне невозможна, так как рано или поздно неравенство $m > n$ будет нарушено. Невозможна и накачка на стыке зон, так как будет появляться недопустимое вхождение цепочки ba. Но накачивать цепочку в зоне символов a можно сколько угодно. Но тут мы должны вспомнить, что по условию леммы о разрастании накачиваемую, среднюю, цепочку можно выбросить.

Рассмотрим тогда слово $a^{n+1}b^n \in L_5, n > 0$.

В предположении регулярности анализируемого языка любое такое слово должно удовлетворять лемме о разрастании. Но тогда, если мы положим $v = a^k, 0 < k \leq n+1$, выбрасывание этой цепочки даст слово $a^{n+1-k}b^n \notin L_5$, так как $n+1-k \leq n$.

Итак, данный язык нерегулярен.

Точно также доказывается нерегулярность языка, где в определении строгое неравенство заменено нестрогим или противоположным неравенством.

6) $L_6 = \{a^m b^n : m \neq n, m, n \geq 0\}$.
Здесь возникает неравенство чисел вхождений букв неопределенного знака.

Решить эту задачу можно так.

Рассмотрим язык $\bar{L}_6 \cap a^* b^*$. Дополнение языка L_6 содержит все слова, в которых буквы могут как угодно перемешиваться, а из слов, в которых строго сначала идет цепочка букв a, а за ней – цепочка букв b, только те, в которых числа вхождений букв равны. Поэтому $\bar{L}_6 \cap a^* b^* = L_1 = \{a^n b^n : n \geq 0\}$ есть нерегулярный язык (см. пример 1). Но тогда и исходный язык L_6 нерегулярен, ибо, если бы он был регулярен, то регулярно было бы его дополнение и записанное выше пересечение, но оно нерегулярно.

Лекция №19

29.11.24

Полезно иметь в виду такое следствие из леммы о разрастании:

Следствие. В любом бесконечном регулярном языке существует последовательность слов, длины которых образуют возрастающую арифметическую прогрессию.

Доказательство. В качестве такой последовательности можно взять последовательность слов $x_n = uv^n w, n \geq 0$ из условия леммы. Разностью прогрессии будет как раз длина средней цепочки.

Это более слабое, чем сама лемма, необходимое условие регулярности.

7) Язык $L_7 = \{a^{n^2} : n \geq 0\}$.

Это язык в однобуквенном алфавите, длины слов которого являются полными квадратами.

Нерегулярность этого языка следует из того известного факта, что последовательность полных квадратов не может быть выстроена в арифметическую прогрессию:

$$(n+k)^2 - n^2 = 2nk + k^2, k \geq 1,$$

то есть разность между любыми двумя квадратами неограниченно возрастает.

Используя то же следствие, можно доказать нерегулярность языка

$L'_7 = \{a^p : p - \text{ простое число}\}$

Рекомендуется доказать самостоятельно.

(Нужно доказать, что невозможна арифметическая прогрессия, членами которой будут только простые числа.)

Продолжим анализ примеров.

Вернуться к примерам (3) и (5) из предыдущей лекции.

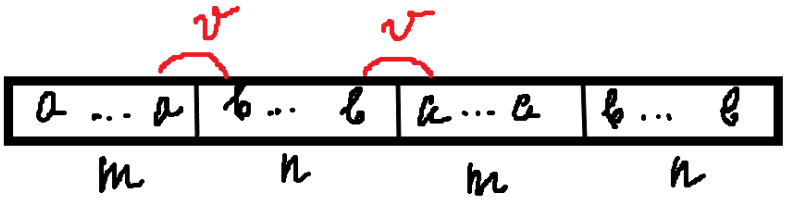
8) Язык двойных слов в алфавите, содержащем не менее 2-х букв:

$L_8 = \{ww : w \in V^*, |V| > 1\}$.

Рассмотрим пересечение

$L_8 \cap a^*b^*a^*b^* = \{a^m b^n a^m b^n : m, n \geq 0\}$

Очевидно, что расположение накачиваемой цепочки v целиком в любой из зон символа a или b невозможно.



Если $v = a^s b^r (0 < s \le m, 0 < r \le n)$, то $v^2 = a^s b^r a^s b^r$, и возникнет второе вхождение цепочки ba , что недопустимо по определению языка.

Если же $v = b^r a^s (0 < s \le m, 0 < r \le n)$ и $v^2 = b^r a^s b^r a^s$, то возникнет третье вхождение цепочки ab , что также недопустимо.

Расположение же накачиваемой цепочки «в обхват» какой-либо зоны невозможно из-за ограничений на длину накачиваемой цепочки: всегда можно выбрать «длины» зон так, чтобы они превосходили предполагаемую константу из леммы о разрастании, которая в силу предположения о регулярности языка где-то фиксирована на числовой прямой.

Следовательно, указанное пересечение нерегулярно, и язык L_7 нерегулярен.

9) $L_9 = \{xcy : x, y \in \{a,b\}^*, c \notin \{a,b\}, |x| \ge |y|\}$

Нужно рассмотреть пересечение

$L_9 \cap a^* c b^* = \{a^m c b^n : m, n \ge 0, m \ge n\}$

Расположение накачиваемой цепочки в зоне символов a отвергается рассмотрением цепочки вида $a^n c b^n, n \ge 0$. В таком случае выбрасывание накачиваемой цепочки приведет к выходу за пределы данного языка.

10) $L_{10} = \{xcy : x, y \in \{a,b\}^*, c \notin \{a,b\}, |x| \ne |y| + k, k > 0\}$

Самостоятельно.

Рекомендуется самостоятельно решить задачи 7.35 в), д), з) из Учебника.

Дополнение. Некоторые более трудные задачи на лемму о разрастании

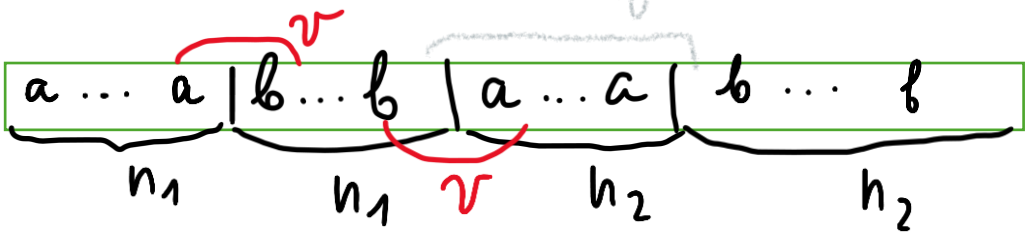
Помимо примеров, разобранных на лекции, рассмотрим некоторые более трудные задачи.

1) Язык $L = L_1^2 L_1^*$, где $L_1 = \{a^n b^n : n > 0\}$.
   Чтобы доказать нерегулярность такого языка, рассмотрим пересечение
   $L_2 = L \cap a^* b^* a^* b^* = \{a^{n_1} b^{n_1} a^{n_2} b^{n_2} : n_1, n_2 > 0\}$.

Заметим, что в силу того, что пустая цепочка не принадлежит языку L_1, каждая цепочка языка L начинается префиксом вида $a^{n_1}b^{n_1}a^{n_2}b^{n_2};n_1,n_2>0$, поэтому пересечение $L\cap a*b*=\varnothing$.

Рассмотрим тогда достаточно длинную цепочку записанного выше пересечения L_2.

На рисунке ниже показаны различные возможные способы расположения накачиваемой цепочки v из леммы о разрастании.



Исключая очевидно невозможное расположение накачиваемой цепочки внутри какой-либо «зоны» (символов a или b), проанализируем накачку на стыках зон.

Если $v=a^kb^l,0<k,l<n_i(i\in\{1,2\})$, получим, что $v^2=a^kb^la^kb^l$, то есть при накачке возникнут новые вхождения цепочки ba, что противоречит структуре цепочек языка L_2, в которых цепочка ba входит ровно один раз.

Если же $v=b^la^k,0<k,l<n_i(i\in\{1,2\})$, то $v^2=b^la^kb^la^k$, и накачка приведет к появлению лишнего вхождения цепочки ab, которая в каждую цепочку языка L_2 входит ровно два раза.

Расположение же накачиваемой «в обхват» зоны (серый цвет на рисунке) невозможно ввиду ограничения на длину накачиваемой цепочки: $|v|\le k_L$. Константа k_L, фиксируемая где-то на числовой прямой в силу предположения о регулярности анализируемого языка, может быть сколь угодно превзойдена выбором параметров n_1 и n_2 так, чтобы $n_1,n_2>k_L$. Эту константу мы, разумеется, знать не можем, но само предположение о регулярности гарантирует существование этой константы как *фиксированной* точки на числовой прямой, которую можно превзойти, задав длины «зон» с учетом того, что длины цепочек анализируемого языка не ограничены сверху.

Итак, язык L_2 нерегулярен, и вместе с ним нерегулярен и исходный язык L.

2) Язык $L=L_1^3L_1^*$, где $L_1=\{a^mb^n:m,n>0,m\ne n\}$.

Поскольку каждая цепочка этого языка начинается префиксом

$$a^{m_1}b^{n_1}a^{m_2}b^{n_2}a^{m_3}b^{n_3};m_i\ne n_i,i\in\{1,2,3\},$$

пересечем его с регулярным языком $(a*b*)^3$. В пересечении получим:

$$L_2=L\cap(a*b*)^3=\{a^{m_1}b^{n_1}a^{m_2}b^{n_2}a^{m_3}b^{n_3};m_i\ne n_i,m_i,n_i>0;i\in\{1,2,3\}\},$$

а чтобы доказать нерегулярность полученного пересечения, рассмотрим язык

$$L_3=\bar{L}_2\cap(a^+b^+)^3=\{a^{n_1}b^{n_1}a^{n_2}b^{n_2}a^{n_3}b^{n_3};n_i>0,i\in\{1,2,3\}\}.$$

Заметим, что выбор позитивной итерации в записанном выше выражении принципиален, так как в дополнение языка L_2 попадут цепочки вида

$a^{m_1}b^{n_1}a^{m_2}b^{n_2}a^{m_3}b^{n_3}(m_i,n_i \geq 0, i \in \{1,2,3\})$, то есть возможны пропуски некоторых зон, и соотношения между числами вхождений символов будет произвольным, так как неравенство $m_i \neq n_i$ выполняется только для двух соседних зон.

Нерегулярность языка L_3 доказывается совершенно аналогично доказательству нерегулярности языка L_2 в предыдущей задаче.

3) В продолжение предыдущего примера рассмотрим язык:

$L = \{a^n b^m a^p : m,n \geq 0, p > n\}$.

Понятно, что сразу применить лемму о разрастании тут нельзя, так как невозможно будет отвергнуть возможность расположения накачиваемой цепочки в зоне символов b.

Образуем пересечение

$L_1 = L \cap a^* b a^* = \{a^n b a^p : n \geq 0, p > n\}$.

Все случаи расположения накачиваемой цепочки легко отвергаются, кроме одного – в правой зоне символов *a*. Но тут используем «прием выбрасывания» накачиваемой цепочки (ведь по лемме о разрастании ее можно не только повторять сколько угодно раз, но можно и выбросить). Тогда рассмотрим цепочку $a^n b a^{n+1}$. Если положить $v = a^k, 0 < k < n+1$ и поместить ее в правую зону символов *a*, то после ее выбрасывания получим цепочку $a^n b a^{n+1-k}$. Так как $n+1-k \leq n$, то такая цепочка уже не будет принадлежать языку L_1, который, следовательно, нерегулярен, а вместе с ним нерегулярен и исходный язык L.

Можно показать, что и этот (исходный) язык удовлетворяет лемме о разрастании.

Действительно, если $m > 0$, то можно положить $v = b$. Если же $m = 0$, то есть цепочка имеет вид $a^n a^{n+k} = a^{2n+k}, k > 0$, то можно положить $v = a^2$. При накачке получим цепочки вида $a^{2(n+m)+k}, m,k \geq 1$, которые принадлежат языку L. Значит, можно принять $k_L = 3$ (очевидно, самая короткая цепочка при $m > 0$ есть ba, а $a \in L$, но $\lambda \notin L$, и цепочку a^2 в качестве исходной при $v = a^2$ взять нельзя).

Самостоятельно: доказать нерегулярность следующих языков:

1) $L = \{w^n : w \in V^*, |V| > 1, n > 1\}$ (n - фиксированное число).
2) $L = \{a^m b^n c^p : m,n,p \geq 0, m = p^2 + 1\}$.
3) Язык L определяется уравнением: $x = ab + axb + x^2$.
4) $L = \{xc^n y : x,y \in \{a,b\}, c \notin \{a,b\}, |x| = |y|, n > 0\}$.