DG

Оглавление

Глава 0

Формальные языки: определения

Основной объект изучения в этом курсе — формальные языки и связанные с ними конструкции.

Определение. *Алфавит* есть конечное множество; будем обозначать его $\Sigma = \{a_1, \ldots, a_N\}$. *Слово* w есть конечная последовательность $w_1 w_2 \ldots w_k$ символов алфавита. К i-ой букве слова w мы будем обращаться $w[i]$; к первой букве слова обращаемся как к $w[1]$ [то есть отсчет начинаем с единицы, а не с нуля]. Длину слова w мы будем обозначать как $|w|$, а $\#_x(w)$ — количество вхождений буквы x в w.

Пример 1. Над алфавитом $\{0, 1\}$ словами являются $w_1 = 0010$ и $w_2 = 1010110$. В слове w_1 ровно одна единица, то есть $\#_1(w_1) = 1$, а $|w_2| = 7$ — во втором слове 7 букв. В обоих словах одинаковая третья буква, то есть $w_1[3] = w_2[3]$. При этом $w_1[5]$ не определена, так как $|w_1| = 4$.

Пустую строчку ϵ определим как слово нулевой длины. Оно является подсловом любого слова.

Определение. Множество всех слов над алфавитом $\Sigma = \{a_1, \ldots, a_N\}$ будем обозначать $\Sigma^* = \{a_1, \ldots, a_N\}^*$; *язык* L есть подмножество Σ^*.

Заметим, что $\varnothing^* = \{\epsilon\}$.
Теперь определим на словах операцию умножения.

Определение. *Конкатенация* двух слов w_1 и w_2 есть просто слово

$$w_1 w_2 := w_1[1] \ldots w_1[|w_1|] w_2[1] \ldots w_2[|w_2|].$$

Пример 2. Для слов $w_1 = ab$ и $w_2 = baa$ имеем $w_1 w_2 = abbaa$ и $w_2 w_1 = baaab$. Заметим, что $w_1 w_2 \neq w_2 w_1$.

Любое непустое слово является конкатенацией своих букв. Операция конкатенации обладает следующими свойствами:

- она ассоциативна: $(ab)c = a(bc)$;
- имеется единица относительно такого умножения: $\epsilon a = a\epsilon = a$;
- $|ab| = |a| + |b|$.

Таким образом, Σ^*, оснащенное конкатенацией, есть *свободный моноид*, то есть полугруппа с единицей, не имеющая соотношений; последнее означает, что никакие два разных слова не равны друг другу. В дальнейшем будем писать $x^n = \underbrace{x \ldots x}_{n \text{ раз}}$. Например, $bbaaaabccccc = b^2 a^4 b c^5$.

Реверс, или *обращение* слова w, определено как $w^R = w[|w|] \ldots w[1]$; иными словами, это слово w, прочитанное наоборот. Слово, совпадающее со своим обращением, будем называть *палиндромом*.

Определение. Пусть $A, B \subset \Sigma^*$. *Объединение* языков $A \cup B$ есть их объединение как множеств. *Произведение* языков AB есть множество всевозможных конкатенаций слов из A и B:

$$AB := \{xy | x \in A,\ y \in B\} \tag{1}$$

Заметим, что $A\varnothing = \varnothing A = \varnothing$, а $A\{\epsilon\} = \{\epsilon\}A = A$. Считаем далее $A^n = \underbrace{A \dots A}_{n \text{ штук}}$. *Итерация* языка A^* есть множество слов, составленных из слов A как из букв:

$$A^* = \{\epsilon\} \cup \bigcup_{i=1}^{\infty} A^i.$$

Соответствующую операцию $*$ мы будем называть *звездой Клини*. Дополнительно мы введем $A^+ = AA^*$. *Обращением* языка A будем называть

$$A^R := \{x^R | x \in A\} \tag{2}$$

Пример 3. Пусть $A = \{\epsilon, b\}$, $B = \{a, ba, b^2a\}$. Тогда $A^2 = \{\epsilon, b, b^2\}$, $BA = \{a, ba, b^2a, ab, bab, b^2ab\}$, $(AB)^R = \{a, ab, ab^2, ba, bab, bab^2\}$. При этом $A = A^R$ и $A^* = \{a\}^*$.

Задача 0.1. Покажите, что B^*A^2 есть множество слов от букв a и b, которые не содержат трех b подряд.

Задача 0.2. Пусть $\Sigma = \{a, b\}$. Существуют ли два разных языка X таких, что $X = aXb + \epsilon$?

Решение. Если бы существовали два таких языка A и B, что $A = aAb + \epsilon$ и $B = aBb + \epsilon$, то в обоих языках нет слов нечетной длины и ровно одно слово четной длины $2n$, а именно a^nb^n. Ведь если $X = aXb + \epsilon$, то $X \cap \Sigma^{2n} = a(X \cap \Sigma^{2n-2})b$, а $X \cap \Sigma^0 = \epsilon$, поэтому если $X \cap \Sigma^{2n-2} = a^{n-1}b^{n-1}$, то $X \cap \Sigma^{2n} = a^nb^n$. Поэтому $A = B$, двух разных решений уравнения не существует. □

Задача 0.3. Проверьте истинность следующих равенств:

(a) $A^*A^* = A^*$;

(b) $A^* = \{\epsilon\} \cup AA^*$;

(c) $(A^*)^* = A^*$.

Задача 0.4. Алфавит содержит хотя бы две буквы. Скажем, что $L \in \Sigma^*$ *треугольный*, если

$$\forall x, y, z \in \Sigma^+,\ xyz \in L \iff yzx, zxy \in L \tag{3}$$

Верно ли, что если L треугольный, то и L^2 треугольный?

Решение. Ответ: нет.

Зафиксируем алфавит $\Sigma = \{a, b\}$ и возьмем язык $L = \{aab, aba, baa\}$. Этот язык является треугольным: все слова можно разбить лишь на слова длины 1, каждое слово присутствует вместе со всеми своими циклическими сдвигами. Между тем, в L^2 есть слово $baaaab$, его можно разбить как $ba^4b = b \cdot a^4 \cdot b$ и циклически переставить множители, тогда полученное слово $a^4b^2 \notin L^2$, ведь $a^3 \notin L$, а слов длины, отличной от 3, в L тоже нет.

Леша Крошнин предложил еще одно решение этой задачи: язык $L = \{ab\}$ в алфавите $\Sigma = \{a, b\}$. L треугольный, так как условие треугольности не выполнено вообще никогда. Между тем, $L^2 = \{abab\}$ очевидно не треугольный, ведь $abab \in L^2$, но $baba \notin L^2$: возьмем $x = a,\ y = ba,\ z = b$. □

Определение. Слово p является *префиксом* слова w, если $\exists s \in \Sigma^*$ такое, что $ps = w$. Мы будем записывать $p \sqsubseteq w$. Аналогично, слово s является *суффиксом* слова w, если $\exists p \in \Sigma^*$ такое, что $ps = w$. Мы будем называть префиксы и суффиксы w *собственными*, если они не совпадают со всем w. *Перехлест* двух слов $w_1, w_2 \in \Sigma^*$ есть максимальный по длине суффикс w_1, являющийся некоторым префиксом w_2. Его будем обозначать как $overlap(w_1, w_2)$.

Задача 0.5. Скажем, что для двух слов $x, y \in \Sigma^*$ максимальное слово $\max(x, y)$ есть максимальное по длине из x и y. Корректно ли определена такая операция? А если $x = overlap(x_1, w), y = overlap(y_1, w)$ для некоторого $w \in \Sigma^*$?

Задача 0.6 (Higman [?koz])**.** Определим на Σ^* отношение квазипорядка: $x \preceq y$, если x получается из y удалением нескольких символов; так, $a^3 \preceq (ac)^3$. Докажите *лемму Хигмана*: любой $L \subset \Sigma^*$ имеет конечное число $\preceq$-минимальных элементов.

Морфизмы есть просто морфизмы моноидов $h : \Sigma^* \to \Gamma^*$, то есть отображения, сохраняющие единицу и мультипликативность:

$$h(xy) = h(x)h(y),\ h(\epsilon) = \epsilon \tag{4}$$

Тогда определим *образ* множества $X \subset \Sigma^*$ как $h(X) := \{y \in \Gamma^* | \exists x \in X\ h(x) = y\}$ и *прообраз* множества $Y \subset \Gamma^*$ как $h^{-1}(Y) := \{x \in \Sigma^* | \exists y \in Y\ h(x) = y\}$.

Задача 0.7. $\Sigma = \{a, b\}$. Определим индуктивно последовательность *слов Фибоначчи* следующим образом: $f_0 = \epsilon$, $f_1 = b$, $f_2 = a$, $f_k = f_{k-1}f_{k-2}$ для всех $k \geqslant 3$. Пусть $\phi : \Sigma^* \to \Sigma^*$ — морфизм, заданный на алфавите $\phi(a) = ab$, $\phi(b) = a$. Покажите, что $\phi^k(a) = f_{k+2}$.

Задача 0.8 (Thue-Morse [?shallit])**.** Рассмотрим два бесконечных двоичных слова X и Y. Первое слово есть $X = \lim\limits_{n \to \mathbb{N}} X_n$, предел заданной индуктивно последовательности $\{X_n\}_{n \in \mathbb{N}}$: $X_0 = 0$, $X_{n+1} = X_n\overline{X_n}$, где $\overline{A}$ получается из A заменой 0 на 1 и наоборот. Y есть бесконечная последовательность $y_0y_1y_2\ldots$, где y_n есть остаток по модулю 2 суммы цифр двоичной записи числа n. Покажите, что $X = Y$. Определим $\mu : \{0, 1\}^* \to \{0, 1\}^*$ следующим образом: $\mu(0) = 01$, $\mu(1) = 10$. Покажите, что в условиях предыдущей задачи $\mu^n(0) = X_n$.

Задача 0.9. Каждое натуральное число однозначно представляется в виде $n = 2^{k_n}(4s_n + t_n)$, где $t_n \in \{1; 3\}$. Пусть $T = t_1t_2\ldots t_n\ldots$, а последовательности слов $\{x_n\}_{n \in \mathbb{N}}$ и $\{y_n\}_{n \in \mathbb{N}}$ определены индуктивно: $x_1 = 1$, $y_1 = 3$, $x_{n+1} = x_n1y_n$, $y_{n+1} = x_n3y_n$. Покажите, что $x_n \sqsubseteq T$ для всех n и $T = \lim\limits_{n \to \infty} x_n$.

Как мы заметили выше, конкатенация не является коммутативной операцией. Оказывается, что существует простой критерий коммутирования двух слов.

Теорема 0.1 (Lyndon, Schützenberger)**.** *Два слова $x, y \in \Sigma^+$ коммутируют тогда и только тогда $\exists z \in \Sigma^+$ такое, что $x, y \in \{w\}^+$.*

Доказательство. Если $x = z^i$ и $y = z^j$ для некоторых $i, j \in \mathbb{N}$ и $z \in \Sigma^+$, то $xy = yx = z^{i+j}$. В обратную сторону утверждение докажем индукцией по $|xy|$. При $|xy| = 2$ имеем $|x| = |y| = 1$, то есть x и y — буквы алфавита, тогда $xy = yx$ влечет $x = y$. Далее, считая утверждение доказанным при $|xy| < n$, докажем его при $|xy| = n$. Если $|x| = |y|$, то $x = y$, поэтому будем считать без ограничения общности, что $|x| < |y|$. КАРТИНКА! В такой ситуации $w \in \Sigma^+$ является одновременно и суффиксом, и префиксом w. Отсюда следует, что $x = wy = yw$; при этом $|yw| < |xy| = n$, поэтому по предположению индукции y и w являются степенями некоторого слова u: $y = u^i$, $w = u^j$ для некоторых $i, j \in \mathbb{N}$ и $u \in \Sigma^+$. Тогда $x = yw = u^{i+j}$ также является степенью слова u. □

Задача 0.10 (Lyndon, Schützenberger [?lyndonschutz])**.** Пусть $x, y, z \in \Sigma^+$. Тогда $xy = yz$ тогда и только существуют $u \in \Sigma^+$, $v \in \Sigma^*$ и $n \geqslant 0$ такое, что $x = uv, y = (uv)^n u, z = vu$.

Задача 0.11. Сформулируйте и докажите аналоги теоремы Линдона-Шютценберже для равенств

(a) $xy = y^R x$;

(b) $xy = y^R z$.

Задача 0.12. **(a)** (Lyndon [?lyndon]) Пусть $x, y, z \in \Sigma^+$. Покажите, что $x^2y^2 = z^2$ титтк $u \in \Sigma^+$ такое, что $x, y \in \{u\}^+,\ z = xy$.

(b) Пусть $x_1^2x_2^2x_3^2 = x_4^2$ для $x_1, x_2, x_3, x_4 \in \Sigma^+$. Обязаны ли хотя бы какие-то два слова x_i и x_j коммутировать?

Задача 0.13. Пусть $x, y \in \Sigma^+$, а $n \geqslant 2$. Докажите, что $(xy)^n = x^ny^n$ титтк $xy = yx$.

Решение. Если x и y коммутируют, то согласно теореме Линдона-Шютценберже $x = w^k$, $y = w^l$ для некоторого $w \in \Sigma^+$ и $k, l \in \mathbb{N}$ и $(w^{k+l})^n = w^{kn}w^{ln}$. Нетривиально доказать утверждение в обратную сторону.

Сразу сократим на x слева и на y справа и будем доказывать следующее утверждение.

Если $(yx)^N = x^Ny^N$ *для* $x, y \in \Sigma^+$ *и* $N \geqslant 1$, *то* $xy = yx$.

Если $|x| = |y|$, то $x = y$ и x коммутирует с y; будем считать без ограничения общности, что $|y| > |x|$. Тут возможны два случая:

- $y \sqsubseteq x^k$ для некоторого $k \leqslant N$;
- $x^N \sqsubseteq y$.

КАРТИНКИ, ДВЕ ШТУКИ!

В первом случае $y = x^{k-1}w$, а $x = wu$, то есть $y = (wu)^{k-1}w$. Сразу после x^k в слове x^Ny^N обязательно идет подслово w как префикс либо x, либо y; в слове $(yx)^N$ сразу после y идет x. Так как $|w| + |u| = |x|$, то $uw = x$, следовательно, $u, w \in \{z\}^+$ для некоторого $z \in \Sigma^+$, а, следовательно, $x, y \in \{z\}^+$.

Во втором случае совершенно аналогично получаем $y = x^Nw = wx^N$, тогда коммутируют w и x^N. Следовательно, $w^i = x^{Nj}$ для некоторых $i, j \in \mathbb{N}$, откуда по теореме Линдона-Шютценберже получаем, что w и x коммутируют, а значит, $xy = yx$.

Случай $|x| > |y|$ доказывается аналогично.

Отметим также, что существует моноидальный «аналог» великой теоремы Ферма [?schenkman], [?shallit]: уравнение $x^ny^m = z^k$ в строчках имеет решение титтк $\exists w \in \Sigma^+$ такое, что $x, y, z \in \{w\}^+$. Пользуясь этим утверждением, можно получить решение задачи в одну строчку: мы решаем систему уравнений

$$\begin{cases} z^n = x^ny^n \\ z = xy \end{cases},$$

согласно первому же уравнению имеем, что $x, y, z \in \{w\}^+$ для некоторого w. □

С помощью формальных языков мы научимся получать некоторые результаты про производящие функции. Напомним, что *производящая функция* последовательности $\{a_n\}_{n\in\mathbb{N}}$ есть $f(z) = \sum\limits_{n\in\mathbb{N}} a_nz^n$, где z — формальная переменная. Последнее означает, что мы не рассматриваем производящие функции как отображения $f : \mathbb{R} \to \mathbb{R}$, а как формальные ряды с соответствующими операциями сложения и умножения

$$\Big(\sum_{n\in\mathbb{N}} a_nz^n\Big) + \Big(\sum_{n\in\mathbb{N}} b_nz^n\Big) = \sum_{n\in\mathbb{N}} (a_n + b_n)z^n;\ \Big(\sum_{n\in\mathbb{N}} a_nz^n\Big)\Big(\sum_{n\in\mathbb{N}} b_nz^n\Big) = \sum_{n\in\mathbb{N}} \Big(\sum_{i\in[0;n]} a_ib_{n-i}\Big)z^n.$$

В некоторых целях, однако, полезно рассмотреть производящие функции как вещественнозначные или комплекснозначные отображения; эту тему подробно рассмотрел Филипп Флажоле [?flajolet].

Производящая функция языка L есть же $f_L(z) = \sum\limits_{n\in\mathbb{N}} |L \cap \Sigma^k|z^k$. Иными словами, это производящая функция последовательности $\{a_n\}_{n\in\mathbb{N}}$, где a_n — число слов длины n в языке L.

Пример 4. Производящая функция языка $L = \{a^2, ab, bab, a^3, aba, b^2a, b^4\} \subset \{a,b\}^*$ равна $f_L(z) = 2z^2 + 4z^3 + z^4$. Для языка Σ^* имеем

$$f_{\Sigma^*}(z) = \sum_{n=0}^{\infty} |\Sigma|^n z^n = \frac{1}{1 - |\Sigma| z}$$

Задача 0.14. Вычислите производящие функции следующих языков:

(a) $\{a^n b^n c^n \mid n \in \mathbb{N}\} \subset \{a,b,c\}^*$;

(b) $\{awb \mid w \in \{a,b,c\}^*\}$;

(c) $Pref(w) = \{p \in \Sigma^* \mid p \sqsubseteq w\}$ [для любого $w \in \Sigma^*$];

(d) $L_k = \{w \in \{a,b\}^* \mid \#_b(w) = k\}$ [для любого $k \in \mathbb{N}$];

(e) $\{w \in \Sigma^* \mid w = w^R\}$.

Задача 0.15. Верно ли, что $f_{L_1+L_2}(z) = f_{L_1}(z) + f_{L_2}(z)$? А верно ли, что $f_{L_1 L_2}(z) = f_{L_1}(z) f_{L_2}(z)$?

Решение. Разумеется, имеет место формула $f_{L_1+L_2}(z) + f_{L_1 \cap L_2}(z) = f_{L_1}(z) + f_{L_2}(z)$, поэтому формула $f_{L_1+L_2}(z) = f_{L_1}(z) + f_{L_2}(z)$ неверна для любых пересекающихся языков. Достаточно взять два совпадающих непустых языка, тогда $f_L(z) = 2f_L(z)$, что очевидно неверно. □

Решение. Ответ крылся в предыдущем пункте: да, верно, что $(L_1 L_2)_n = \cup_{k \in [0;n]} (L_1)_k (L_2)_{n-k}$, где $(L)_k = \{w \in L \| w| = k\}$ — множество слов языка L длины k. Но эти множества могут пересекаться: например, если $L_1 = \{\epsilon, a\}$, а $L_2 = \{a, a^2\}$, то $a^2 = \epsilon \cdot a^2 = a \cdot a$. Заметим, что здесь получается $L_1 L_2 = \{a, a^2, a^3\}$, и

$$f_{L_1} f_{L_2}(z) = (1+z)(z+z^2) = z + 2z^2 + z^3 \neq z + z^2 + z^3 = f_{L_1 L_2}(z)$$

Альтернативно можно было поступить как Тагир: «А, нафиг все. Пусть $L = a^*$ — множество всех слов от одной буквы, тогда $L^2 = L$, и $f_L^2 = f_L$, откуда $f_L = 0$ или $f_L = 1$. Что совершенно точно не может быть правдой» □

Задача 0.16. Пусть $\Sigma = \{a_1, \ldots, a_k\}$. Скажем, что *Парик-производящая функция* — это

$$F_L(z_1, \ldots, z_k) = \sum_{(n_1, \ldots n_k) \in \mathbb{N}^k} N(n_1, \ldots n_k) z_1^{n_1} \ldots z_k^{n_k},$$

где $N(n_1, \ldots n_k) = |\{w \in L | \#_{a_1}(w) = n_1, \ldots \#_{a_k}(w) = n_k\}|$. Пусть $h : \Sigma^* \to \Gamma^*$ — морфизм. Чему равна $F_{h(L)}(z_1, \ldots, z_k)$?

Введем также $\mathbb{Z}\langle\langle\Sigma\rangle\rangle = \{f : \Sigma^* \to \mathbb{Z}\}$. Мы будем работать с этими отображениями $f : \Sigma^* \to \mathbb{Z}$ как с некоммутативными производящими функциями, то есть с суммами вида $\sum_{w \in \Sigma^*} f_w w$, где $f_w \in \mathbb{Z}$. Операции сложения и умножения вводятся на некоммутативных производящих функциях следующим образом:

$$\Big(\sum_{w \in \Sigma^*} a_w w\Big) + \Big(\sum_{w \in \Sigma^*} b_w w\Big) = \sum_{w \in \Sigma^*} (a_w + b_w) w; \quad \Big(\sum_{w \in \Sigma^*} a_w w\Big)\Big(\sum_{w \in \Sigma^*} a_w w\Big) = \sum_{w \in \Sigma^*} \Big(\sum_{u_1 u_2 = w} a_{u_1} b_{u_2}\Big) w$$

Остальные операции будем в дальнейшем определять через сложение и умножение; например, $\exp(\sum_{w \in \Sigma^*} a_w w) = \sum \frac{1}{n!} \big(\sum_{w \in \Sigma^*} a_w w\big)^n$.

Задача 0.17. **(a)** Пусть $f_1, f_2 \in \mathbb{Z}\langle\langle\Sigma\rangle\rangle$. Покажите, что $\exp(f_1)\exp(f_2) = \exp(f_1 + f_2)$ титтк $f_1 f_2 = f_2 f_1$.

(b) Пусть $f_1, f_2 \in \mathbb{Z}\langle\langle\Sigma\rangle\rangle$ таковы, что $f_1 f_2 = f_2 f_1$. Существуют ли $C_1, C_2 \in \mathbb{Z}$ и $f \in \mathbb{Z}\langle\langle\Sigma\rangle\rangle$, что $\frac{f_1}{C_1} = f^{k_1}$, $\frac{f_2}{C_2} = f^{k_2}$ для некоторых $k_1, k_2 \in \mathbb{N}$?

Глава 1

Детерминированные конечные автоматы

Предмет нашего изучения — модели вычислительных машин, их возможности, свойства и применения. Машины, которые мы будем рассматривать, решают *задачу о принадлежности* некоторому языку, то есть получают на вход слово, составленное из букв некоторого алфавита, и выдают «да» или «нет» в зависимости от того, принадлежит ли поданное на вход слово языку.

Простейшая из таких машин — конечный автомат, машина, множество состояний которой конечно.

Определение. *Детерминированный конечный автомат* — набор $(Q, \Sigma, Start, Final, T)$, где

- Q — конечное множество состояний,
- Σ — алфавит, конечное множество символов,
- $Start \in Q$ — стартовое состояние, ровно одно,
- $Final \subseteq Q$ — подмножество финальных состояний,
- T — таблица переходов, отображение $Q \times \Sigma \to Q$:

$$T(q_i, x) = q_j \tag{1.1}$$

Мы будем писать $q_i \xrightarrow{x} q_j$, если $T(q_i, x) = q_j$.

Конфигурация ДКА есть элемент $Q \times \Sigma^*$. Таблица переходов T задет отображение пространства конфигураций:

$$\forall q \in Q, x \in \Sigma, w \in \Sigma^* \qquad (q, xw) \mapsto (T(q, x), w)$$

Тогда индуктивно продолжим таблицу переходов T до отображения $\widetilde{T} : Q \times \Sigma^* \to Q$:

$$\forall q \in Q, x \in \Sigma, w \in \Sigma^* \qquad \widetilde{T}(q, \epsilon) = q,\ \widetilde{T}(q, wx) = T(\widetilde{T}(q, w), x) \tag{1.2}$$

Фактически $\widetilde{T}(q, w)$ есть состояние, в котором автомат окажется, стартовав из состояния q и обработав слово w. Это можно проверить индукцией по $|w|$, доказательство предоставляется читателю. Для краткости будем писать $q \xrightarrow{w} s$, если $\widetilde{T}(q, w) = s$. Слово $\omega \in \Sigma^*$ *принимается* автоматом A, если $\widetilde{T}(Start, w) \in Final$, то есть при переходах по символам этого слова по таблице переходов $T(A)$ автомат A остановится в одном из финальных состояний. *Язык автомата* $L(A)$ — множество всех слов, принимаемых автоматом:

$$L(A) = \{w \in \Sigma^* \mid \widetilde{T}_A(Start, w) \in Final\}$$

Язык детерминированного конечного автомата можно воспринимать как путей из точки A в множество точек B в конечном помеченном ориентированном графе. Граф здесь задается с помощью Q и T: вершины графа — это множества состояний, а ребра имеют вид $(q, T(q, x))$ для любой буквы x. Отдельно мы будем помечать стартовые и финальные состояния; такой граф мы будем называть *диаграммой Мура*.

Определение. Язык $L \subset \Sigma^*$ называется *регулярным*, если $L = L(Aut)$ для некоторого детерминированного конечного автомата Aut.

Данный в первом определении автомат *полный*, то есть по любой букве из любого состояния есть переход в состояние. Можно было определять неполный автомат, где формально таблица переходов будет не функцией (определенной на любой паре $(q, x) \in Q \times \Sigma$), а отношением (множеством троек $(q_1, x, q_2) \in Q \times \Sigma \times Q$, считается, что по букве x совершается переход $q_i \to q_j$). Такой автомат может не прочитать слово до конца и просто «сломаться» где-то в середине слова. Нам удобнее рассматривать полные автоматы и считать, что любое слово будет прочитано до конца (причина станет очевидной позднее).

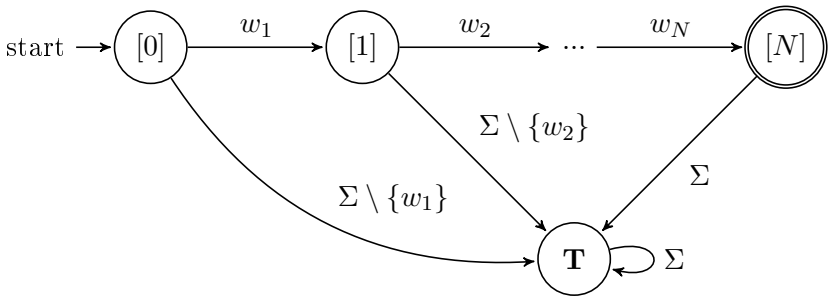
Задача 1.1. Дайте точное определение неполного ДКА, его конфигурации и принимаемого языка. Покажите, что для любого неполного ДКА A можно построить (полный) ДКА $\widehat{A}$ такой, что $L(A) = L(\widehat{A})$. Иными словами, докажите, что неполные ДКА принимают те же языки, что и полные.

Будем говорить, что состояние $q_2 \in Q$ *достижимо из состояния* q_1, если существует $w_{q_1,q_2} \in \Sigma^*$ такое, что $\widetilde{T}(q_1, w_{q_1,q_2}) = q_2$.

Задача 1.2. Предложите алгоритм, позволяющий найти по автомату множество всех состояний, достижимых из стартового состояния.

Мы будем дальше по умолчанию считать, что все состояния достижимы из стартового.

Пример 5. Язык $\{w\} \subset \Sigma^*$, состоящий из единственного слова $w = w_1 \dots w_N$ (для $w_i \in \Sigma$), является регулярным. Действительно, можно построить следующий автомат A:



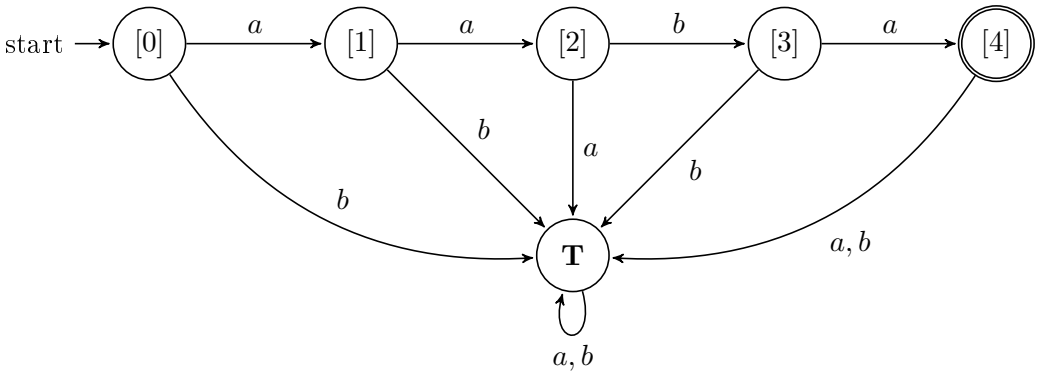
Формально, $A = (Q, \Sigma, Start, Final, T)$

- $Q = \{[i] | i \in [0; N]\} \cup \{\mathbf{T}\}$;
- Σ — данный в условии алфавит;
- $Start = [0]$ — начальное состояние;
- $Final = \{[N]\}$ — финальное состояние;
- T — таблица переходов вида

$$[i] \xrightarrow{w_{i+1}} [i+1];\ \forall x \in \Sigma \setminus \{w_{i+1}\}\ [i] \xrightarrow{x} [\mathbf{T}];\ \forall x \in \Sigma\ [T] \xrightarrow{x} [T] \tag{1.3}$$

Покажем, что этот автомат принимает только слово $w = w_1 \dots w_N$. Действительно, $w \in L(A)$, так как $\widetilde{T}([0], w) = ([N], \epsilon)$; любое же другое слово $u \neq \epsilon$ отличается от w в некоторой позиции, скажем, что i-ая позиция — первая, в которой есть различие, тогда $\widetilde{T}([0], u) = \widetilde{T}([i], u[i]u[i+1] \dots u[|u|]) = \mathbf{T}$.

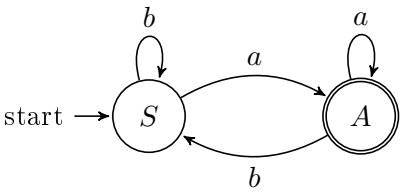
К примеру, для слова $a^2ba \in \{a, b\}^*$ имеем вот такой автомат:



На слове a^3b данный автомат совершает переходы $[0] \xrightarrow{a} [1] \xrightarrow{a} [2] \xrightarrow{a} \mathbf{T} \xrightarrow{b} \mathbf{T}$ и, таким образом, не принимает его.

Замечание. Впоследствии мы установим, что любой конечный язык является языком некоторого автомата. Такая конструкция не обобщается на бесконечные языки (получится бесконечный автомат!), и бесконечные языки, принимаемые автоматами, обладают очень красивыми и сильными свойствами, о которых мы поговорим позднее.

Пример 6. Язык всех слов из букв $\{a, b\}$, оканчивающихся на a, также является языком детерминированного конечного автомата Aut:



Формально говоря, $A = (Q, \Sigma, Start, Final, T)$, где

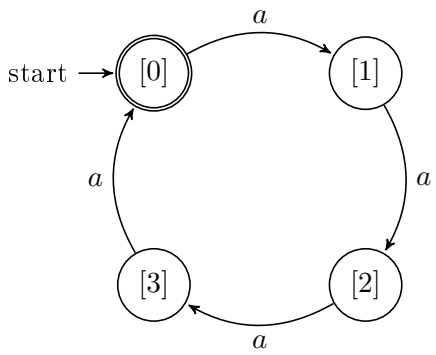
- $Q = \{S, A\}$;
- $\Sigma = \{a, b\}$;
- $Start = S$;
- $Final = \{A\}$;
- T — таблица переходов вида

$$S \xrightarrow{b} S;\ S \xrightarrow{a} A;\ A \xrightarrow{b} S;\ A \xrightarrow{a} A \tag{1.4}$$

На слове a^3ba данный автомат совершает переходы $S \xrightarrow{a} A \xrightarrow{a} A \xrightarrow{a} A \xrightarrow{b} S \xrightarrow{a} A$, а на слове a^2b^2ab — переходы $S \xrightarrow{a} A \xrightarrow{a} A \xrightarrow{b} S \xrightarrow{b} S \xrightarrow{a} A \xrightarrow{b} S$.

Покажем, что построенный автомат принимает язык слов, которые заканчиваются на a. Если слово заканчивается на a, то автомат завершает работу в состоянии a: $T(S, a) = T(A, a) = A$. Если слово заканчивается на b, то так как $T(S, b) = T(A, b) = S$, то оно не принимается автоматом: S не является финальынм состоянием.

Пример 7. Язык всех слов на алфавитом $\{a\}$, длина которых делится на 4, также является регулярным:



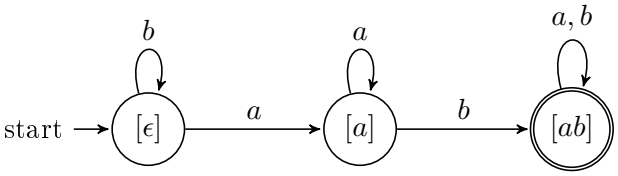
Формально говоря, $A = (Q, \Sigma, Start, Final, T)$, где

- $Q = \{[i] | i \in \mathbb{Z}/4\mathbb{Z}\}$;
- $\Sigma = \{a\}$;
- $Start = [0]$;
- $Final = \{[0]\}$;
- T — таблица переходов вида

$$\forall i \in \mathbb{Z}/4\mathbb{Z} \qquad [i] \xrightarrow{a} [i+1] \tag{1.5}$$

Напомним, что $\mathbb{Z}/4\mathbb{Z}$ — группа остатков по модулю 4, и $i+1$ в таблице переходов воспринимается как элемент этой группы. Тогда $\widetilde{T}([0], a^n) = [i]$ титтк $n \equiv i \bmod 4$: заметим, что $\widetilde{T}([0], \epsilon) = [0]$, а $T(\widetilde{T}([0], a^n), a) = T([n \bmod 4], a) = [(n+1) \bmod 4]$. Поэтому $\widetilde{T}([0], a^m) = [0]$ титтк $m \equiv 0 \bmod 4$, следовательно, язык этого автомата есть $\{a^{4k} | k \in \mathbb{N}\}$.

Пример 8. Язык всех слов из букв $\{a, b\}$, содержащих подслово ab, также является языком детерминированного конечного автомата Aut:



Формально говоря, $A = (Q, \Sigma, Start, Final, T)$, где

- $Q = \{[p] \mid p \sqsubseteq ab\}$ — каждое состояние помечено префиксом искомого слова;
- $\Sigma = \{a, b\}$;
- $Start = [\epsilon]$ — пустой префикс;
- $Final = \{[ab]\}$;
- T — таблица переходов вида

$$[\epsilon] \xrightarrow{b} [\epsilon];\ [\epsilon] \xrightarrow{a} [a];\ [a] \xrightarrow{b} [ab];\ [a] \xrightarrow{a} [a];\ [ab] \xrightarrow{\Sigma} [ab] \tag{1.6}$$

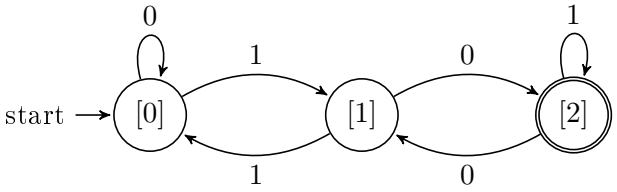
На слове a^3ba данный автомат совершает переходы $[\epsilon] \xrightarrow{a} [a] \xrightarrow{a} [a] \xrightarrow{a} [a] \xrightarrow{b} [ab] \xrightarrow{a} [ab]$.

Покажем, что построенный автомат принимает те и только те слова, которые содержат ab. Если слово содержит ab, то оно имеет вид w_1abw_2 автомат завершает работу в состоянии $[ab]$:

$$\widetilde{T}([\epsilon], ab) = T([a], b) = [ab]; \quad \widetilde{T}([a], ab) = T([a], b) = [ab]; \quad \widetilde{T}([ab], ab) = T([ab], b) = [ab],$$

то есть из любого состояния Aut попадет по слову ab в финальное состояние и больше не покинет его [все переходы из финального состояния ведут в него же]. Если слово w принимается Aut, то рассмотрим самую левую букву $w[l]$ этого слова, по которой Aut, обрабатывая это слово, попал в $[ab]$. Впервые попасть в $[ab]$ можно только из $[a]$, поэтому $[a] \xrightarrow{w[l]} [ab]$, откуда согласно таблице переходов $w[l] = b$. Попасть в состояние $[a]$ можно только по букве a [из состояний $[\epsilon]$ и $[a]$], таким образом, $w[l-1] = a$ и w содержит подслово ab.

Пример 9. Пусть $\Sigma = \{0, 1\}$. Рассмотрим следующий детерминированный конечный автомат:



Формально $A = (Q, \Sigma, Start, Final, T)$, где

- $Q = \{[0], [1], [2]\}$;
- $\Sigma = \{0, 1\}$;
- $Start = [0]$;
- $Final = \{[2]\}$;
- T — таблица переходов вида

$$\forall i \in \{0, 1, 2\},\ x \in \{0, 1\} \qquad [i] \xrightarrow{x} [(2i + x) \bmod 3] \tag{1.7}$$

Действительно, по модулю 3 имеем $2 \cdot 0 + 0 = 0$, $2 \cdot 0 + 1 = 1$, $2 \cdot 1 + 0 = 2$, $2 \cdot 1 + 1 = 0$, $2 \cdot 2 + 0 = 1$, $2 \cdot 2 + 1 = 2$, что согласуется с таблицей переходов

$$[0] \xrightarrow{0} [0]; [0] \xrightarrow{1} [1]; [1] \xrightarrow{0} [2]; [1] \xrightarrow{1} [0]; [2] \xrightarrow{0} [1]; [2] \xrightarrow{1} [2]$$

Заметим, что $11 \in L(A)$, так как $[0] \xrightarrow{1} [1] \xrightarrow{1} [2]$, а $00100 \notin L(A)$, так как

$$[0] \xrightarrow{0} [0] \xrightarrow{0} [0] \xrightarrow{1} [1] \xrightarrow{0} [2] \xrightarrow{0} [1].$$

Пусть $z(\cdot) : \{0, 1\}^* \to \{0, 1\}^*$ — функция, отбрасывающая передние нули, то есть слову w она сопоставляет его суффикс, начинающийся с самой левой единицы. Докажем теперь, что $L(A)$ состоит из слов w таких, что $z(w)$ — двоичная запись числа, дающего остаток 2 по модулю 3. Для начала заметим, что $w \in L(A) \iff 0^k w \in L(A)$ для любого k, так как $\widetilde{T}([0], 0^k w) = \widetilde{T}([0], w)$; поэтому можем считать, что w само по себе является двоичной кодировкой некоторого целого неотрицательного числа $x(w)$. Покажем индукцией по $|w|$, что $\widetilde{T}([0], w) = [x(w) \bmod 3]$. Для слов длины утверждение очевидно: для слова w длины 1 имеем

$$\widetilde{T}([0], w) = T([0], w) = \begin{cases} 0, w = 0 \\ 1, w = 1 \end{cases}$$

Пусть по предположению индукции для любого слова w длины n верно $\widetilde{T}([0], w) = [x(w) \bmod 3]$. Слова $w0$ и $w1$ являются двоичными записями чисел $2x$ и $2x+1$ соответственно. Согласно таблице переходов $[x(w) \bmod 3] \xrightarrow{0} [2x(w) \bmod 3]$ и $[x(w) \bmod 3] \xrightarrow{1} [2x(w) + 1 \bmod 3]$, по предположению $\widetilde{T}([0], w) = [x(w) \bmod 3]$, следовательно,

$$\widetilde{T}([0], w0) = T([x(w) \bmod 3], 1) = [2(x(w) \bmod 3) \bmod 3] = [2x(w) \bmod 3] = [x(w0) \bmod 3],$$
$$\widetilde{T}([0], w1) = T([x(w) \bmod 3], 1) = [2(x(w) \bmod 3) + 1 \bmod 3] = [(2x(w) + 1) \bmod 3] = [x(w1) \bmod 3],$$

тем самым индукционный переход доказан. Таким образом, $\widetilde{T}([0], w) = [2]$ титтк $z(w)$ кодирует число, равное 2 по модулю 3.

Задача 1.3. Укажите, как перестроить вышеуказанный автомат, чтобы он принимал язык двоичных записей чисел вида $\{3k+2 \mid k \in \mathbb{N}\} \subset \mathbb{N}$ *без ведущих нулей.*

Задача 1.4. $\Sigma = \{a\}$. Постройте детерминированные конечные автоматы, принимающие языки

(a) $\{a^k | k \vdots N\}$ для любого $N \in \mathbb{N}$;

(b) $\{a^{100+3k} | k \in \mathbb{N}\}$;

(c) $\{a^{A+Bk} | k \in \mathbb{N}\}$ для любых $A, B \in \mathbb{N}$.

Задача 1.5. $\Sigma = \{a, b\}$. Постройте детерминированные конечные автоматы, принимающие язык слов, в которых

(a) на 2 месте от конца стоит b;

(b) на 3 месте от конца стоит b;

(c) на k месте от конца стоит b;

(d) есть суффикс вида abw, $w \in \Sigma^2$, то есть $|w| = 2$.

Задача 1.6. $\Sigma = \{a, b\}$. Постройте детерминированные конечные автоматы, принимающие следующие языки:

(a) $\{\omega | \#_a(\omega) + 2\#_b(\omega) = 0 \bmod 5\}$;

(b) $\{\omega | 3\#_a(\omega) + 2\#_b(\omega) = 0 \bmod 6\}$;

(c) $\{\omega | 3\#_a(\omega) + 6\#_b(\omega) = 5 \bmod 9\}$;

(d) $\{\omega | A\#_a(\omega) + B\#_b(\omega) = k \bmod N\}$ для любых $A, B, k, N \in \mathbb{N}$.

Задача 1.7. $\Sigma = \{a, b\}$. Постройте детерминированные конечные автоматы, принимающие языки слов, содержащих

(a) ровно три буквы b;

(b) хотя бы две буквы a и две буквы b;

(c) подслово aba;

(d) подслово aba четное число раз;

(e) подслова $(ab)^2b$ и ab^2.

Задача 1.8. $\Sigma = \{0, 1\}$. Постройте детерминированные конечные автоматы, принимающие языки бинарных записей чисел вида

(a) $\{2 + 5k | k \in \mathbb{N}\}$;

(b) $\{1 + 4k | k \in \mathbb{N}\}$;

(c) $\{A + Bk | k \in \mathbb{N}\}$ для любых $A, B \in \mathbb{N}$.

Задача 1.9. $\Sigma = \{0, 1, 2\}$. Постройте детерминированные конечные автоматы, принимающие троичные записи множеств, указанных в прошлой задаче.

Задача 1.10. Пусть $f_i(w_1, w_2) : \Sigma^* \times \Sigma^* \to \mathbb{N}$ определена следующим образом:

$$f_i(w_1, w_2) = \begin{cases} 0, & \text{в обоих словах существует } i\text{-ая буква и } w_1[i] = w_2[i], \text{ либо в обоих словах ее нет} \\ 1, & \text{в противном случае} \end{cases} \tag{1.8}$$

Теперь определим $d(w_1, w_2) = \sum\limits_{i=0}^{\infty} 2^{-i} f_i(w_1, w_2)$. Пусть $\Sigma = \{a, b\}$. Постройте ДКА, принимающий язык

$$L(w) = \{u \in \Sigma^* | d(u, w) \leqslant 0.1\}. \tag{1.9}$$

Задача 1.11. Пусть L — регулярный. Покажите, что язык

$$nL = \{\underbrace{a_1 \dots a_1}_{n \text{ раз}} \underbrace{a_2 \dots a_2}_{n \text{ раз}} \dots \underbrace{a_k \dots a_k}_{n \text{ раз}} | a_1 a_2 \dots a_k \in L\}$$

является регулярным для любого $n \in \mathbb{N}$.

Задача 1.12. Скажем, что $w \in \Sigma^+$ является *словом де Брюйна порядка* k, если любое слово $u \in \Sigma^k$ длины k содержится в w ровно однажды. Например, ab — слово де Брюйна порядка 1 над алфавитом $\{a, b\}$. Покажите, что для любого $k \geqslant 2$ существует слово де Брюйна порядка k.

Задача 1.13. $\Sigma = \{a, b\}$. Постройте детерминированный конечный автомат, принимающие язык слов w, для которых $\forall i \in [1; n]$ верно:

- a^i является подсловом w;
- либо b^i не является подсловом w, либо самое правое вхождение a^i находится справа от самого правого вхождения b^i.

Регулярные подмножества $\{a\}^*$

Регулярные языки над однобуквенным алфавитом легко классифицировать.

Задача 1.14. Докажите, что язык $A \subset (a)^*$ регулярен титтк множество $Degs_A = \{m | a^m \in A\}$ является *асимптотически периодическим*, то есть существуют $n, p \in \mathbb{N}$ так что $\forall m \geqslant n$ $m \in Degs_A \iff m + p \in Degs_A$.

Задача 1.15. Пусть $A \subset \{a\}^*$ — произвольный язык.

(a) Покажите, что A^* регулярен.

(b) Более того, $A^* = \{a^{np}\}_{n \in \mathbb{N}} \backslash G$, где p — наименьший общий делитель $Degs_A$, а G — некоторое конечное множество.

Задача 1.16. **(a)** Докажите, что $\{a^{x^2+y^2} | x, y \in \mathbb{Z}\} \subset \{a\}^*$ не может быть языком никакого ДКА.

(b) Докажите, что $\{a^{x^2-xy+y^2} | x, y \in \mathbb{Z}\} \subset \{a\}^*$ не может быть языком никакого ДКА.

Решение. Есть и альтернативное решение в алфавите из одной буквы. Пусть $Q = \{\epsilon\} \cup \{a^{n^2}\}_{n \in \mathbb{N}}$ — язык неотрицательных квадратов. В силу теоремы о четырех квадратов любое целое неотриицательное число представляется в виде суммы четырех квадратов целых неотрицательных чисел, поэтому $Q^4 = a^*$. Возьмем

$$L = Q^2 = \{a^{n_1^2+n_2^2} | n_i \in \mathbb{N} \cup \{0\}\} \tag{1.10}$$

тогда $L^2 = a^*$. Докажем, что этот язык не может распознаваться никаким ДКА. В силу задачи 3 листка 1А для каждого такого языка существуют N, p так, что $\forall m \geqslant N$ $a^m \in L \iff a^{m+p} \in L$:

p — длина достижимого из старта цикла в графе автомата. Покажем, что p делится на любое простое число π вида $4k+3$. В самом деле, вне зависимости от N существует $4^{\lfloor \log_4 N \rfloor + 1}$ — квадрат, больший N и не делящийся на π; если $p \not\vdots \pi$, то существует некоторое $l \in \mathbb{N}$ такое, что $4^{\lfloor \log_4 N \rfloor + 1} + lp \vdots \pi$, но при этом $4^{\lfloor \log_4 N \rfloor + 1} + lp \not\vdots \pi^2$. Однако $a^{4^{\lfloor \log_4 N \rfloor + 1} + lp} \notin L$, так как число, содержащее простые вида $4k+3$ в нечетной степени, не может представляться как сумма двух квадратов, следовательно, $p \vdots \pi$. Это верно для всех простых $\pi = 4k+3$; однако же простых чисел такого вида бесконечно много (почему?), значит, существует простое число вида $4k+3$, превышающее длину нашего цикла, и, следовательно, не являющееся делителем. Значит, L не может распознаваться никаким ДКА. □

Решение. Эта задача имеет различные решения, приведу известное мне наиболее тривиальное. Все a^{2^k} лежат в языке сумм квадратов: либо 2^{2k} само является квадратом, либо $2^{2k+1} = 2^{2k} + 2^{2k}$. Если бы язык сумм двух квадратов был бы регулярен, то для некоторых $k < m$ два слова a^{2^k} и a^{2^m} будут эквивалентны по Майхилл-Нероду, тогда им же будут эквивалентны все $a^{2^k + l(2^m - 2^k)}$ для $l \in \mathbb{N}$, а тогда $2^k + l(2^m - 2^k)$ являются суммами двух квадратов целых чисел для любых $l \in \mathbb{N}$. Таким образом, для $l = 2$ существуют $A, B \in \mathbb{Z}$ такие, что

$$A^2 + B^2 = 2^k + 2 \cdot (2^m - 2^k) = 2^k(1 + 2 \cdot (2^{m-k} - 1)) = 2^k \cdot M$$

где $M = 3 \bmod 4$. Если $A^2 + B^2 = 2N$, то $(\frac{A+B}{2})^2 + (\frac{A-B}{2})^2 = N$, притом A и B имеют одинаковую четность, поэтому оба $\frac{A \pm B}{2} \in \mathbb{Z}$. Таким образом, если $2^k M$ представимо в виде суммы двух квадратов, то и M представимо, что, разумеется, не может быть правдой: сумма двух квадратов не может давать остаток 3 по модулю 4. □

Как следствие, множества $\{x^2 + y^2 | x, y \in \mathbb{Z}\}$ и $\{x^2 - xy + y^2 | x, y \in \mathbb{Z}\}$ не могут быть представлены в виде конечного объединения арифметических прогрессий.

Автоматы, различающие слова

Скажем, что слова $w_1, w_2 \in \Sigma^*$ *различаются* детерминированным конечным автоматом Aut, если Aut принимает ровно одно из этих двух слов.

Задача 1.17. Покажите, что два языка $L_1 = \{\epsilon, 000, 011, 111\}$ и $L_2 = \{0001\}$ не могут различаться полным ДКА, имеющим всего два состояния.

Задача 1.18. Пусть слова $w_1, w_2 \in \{a\}^*$ имеют длины N и $2N$ соответственно. Верно ли, что любой ДКА, принимающий w_1 и не принимающий w_2, имеет хотя бы $\Omega(\log N)$ состояний?

В следующих двух задачах мы попытаемся получить асимптотику числа состояний детерминированного автомата, различающего два слова. Некоторое знание теории чисел может оказаться полезным.

Задача 1.19. Даны слова $w_1, w_2 \in \{a\}^*$ разных длин, не превышающих n.

(a) Покажите, что они различаются некоторым ДКА, имеющим $O(\log(n))$ состояний.

(b) ([?demaine]) Покажите, что существует бесконечно много пар слов $w_1 \neq w_2 \in \{a\}^*$ таких, что любой различающий их ДКА имеет $\Theta(\log(n))$ состояний.

Задача 1.20 ([?robson])**.** Даны два слова $w_1 \neq w_2 \in \{a, b\}^*$ одинаковой длины n. Будем говорить, что слово $w \in \Sigma^*$ *имеет период длины* p, если $w[i] = w[p + i]$ для всех $i \in [1; |w| - p]$; скажем также, что w *периодично*, если оно имеет период длины $p \leqslant \frac{|w|}{2}$.

(a) Если wa — периодическое слово, то wb не может быть периодическим.

(b) Если для любых $\alpha < 1$ и $w \in \{a, b\}^*$ подслово $w[i] \dots w[i + l - 1]$ непериодично и $l \leqslant |w|^\alpha$, то существует $j \leqslant \mathrm{const}_\alpha \frac{n \log n}{l}$ такое, что

$$\forall k \neq i,\ k \equiv i \bmod j \quad w[k] \dots w[k + l - 1] \neq w[i] \dots w[i + l - 1]$$

(c) Слова w_1 и w_2 различаются некоторым ДКА, имеющим $O(\sqrt{n \log(n)})$ состояний.

Объединение и пересечение

Утверждение 1.1. *Если $A, B \subset \Sigma^*$ являются регулярными, то и $A \cap B$ регулярен.*

Доказательство. Даны два ДКА $Aut_A = (Q_A, \Sigma, Start_A, Final_A, T_A)$ и $Aut_B = (Q_B, \Sigma, Start_B, Final_B, T_B)$, принимающие языки A и B, построим автоматы, принимающие $A \cap B$ и $A \cup B$.

Сначала рассмотрим $Aut = (Q_A \times Q_B, \Sigma, (Start_A, Start_B), Final_A \times Final_B, T)$, где

$$\forall q_A \in Q_A, q_B \in Q_B, x \in \Sigma \; T((q_A, q_B), x) = (T_A(q_A, x), T_B(q_B, x)) \tag{1.11}$$

(PICTURE) Тогда индукцией по $|w|$ мы можем убедиться, что

$$(Start_A, Start_B) \xrightarrow{w}_{Aut} (q_1, q_2) \iff Start_A \xrightarrow{w}_{Aut_A} q_1 \wedge Start_B \xrightarrow{w}_{Aut_B} q_2 \tag{1.12}$$

Действительно, для пустого слова это утверждение очевидно; если оно верно для слова w, то будет верно и для wx, $x \in \Sigma$:

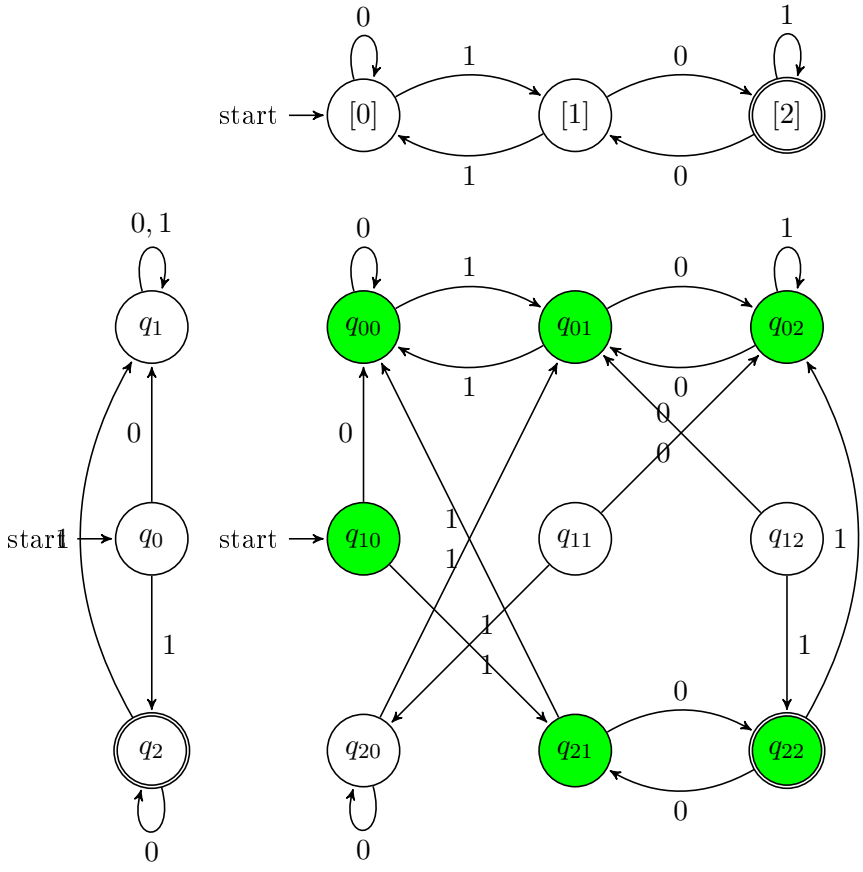
$$(Start_A, Start_B) \xrightarrow{w}_{Aut} (\widetilde{T}_A(Start_A, w), \widetilde{T}_B(Start_B, w)) \xrightarrow{x}_{Aut} (T_A(\widetilde{T}_A(Start_A, w), x)$$
$$T_B(\widetilde{T}_B(Start_B, w), x)) = (\widetilde{T}_A(Start_A, wx), \widetilde{T}_B(Start_B, wx))$$

Тогда w принимается Aut титтк оно принимается и Aut_A, и Aut_B:

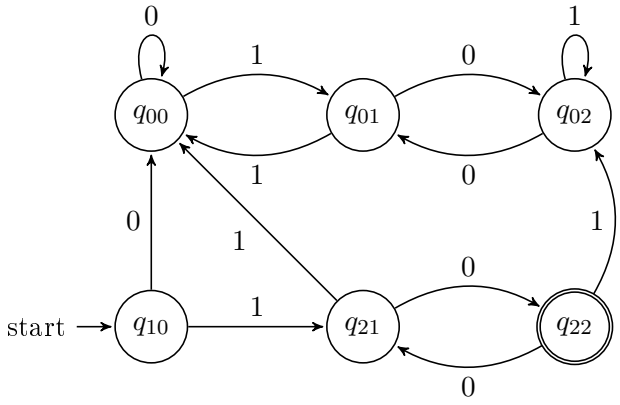
$$w \in L(Aut) \iff \widetilde{T}((Start_A, Start_B), w) \in Final_A \times Final_B \iff$$
$$\widetilde{T}_A(Start_A, w) \in Final_A \wedge \widetilde{T}_B(Start_B, w) \in Final_B \iff w \in A \wedge w \in B.$$

Следовательно, язык построенного автомата есть $A \cap B$. □

Пример 10. Пусть A — язык двоичных записи чисел, дающих остаток 2 по модулю 3 [с ведущими нулями], $B = \{1(0)^k \mid k \in \mathbb{N}\} \subset \{0,1\}^*$. Применим конструкцию из доказательства для построения ДКА, принимающего $A \cap B$.

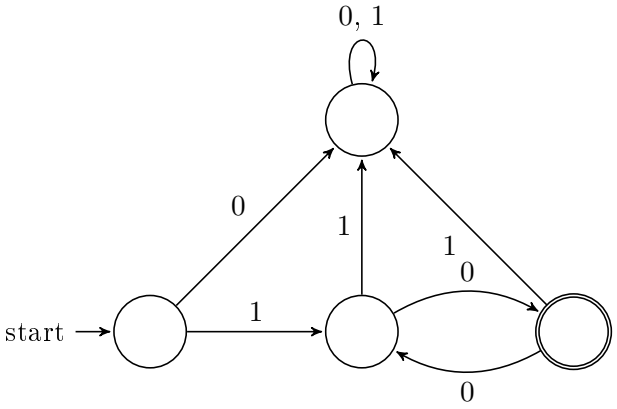


Зеленым цветом выделены состояния, достижимые из старта. Убрав остальные состояния, получим следующий автомат:



Можно ли построить более простой детерминированный автомат для $A \cap B$ из предыдущего пункта?

Заметим также, что $A \cap B$ в примере выше может быть принят более простым автоматом:



Дело в том, что B — множество двоичных записей степеней двойки, так как $2^k = 1\overbrace{0\ldots0}^{k}{}_2$; $2^k = 2 \bmod 3$ титтк k нечетно. Таким образом, $A \cap B = \{1(0)^{2k+1} \mid k \in \mathbb{N}\}$. Несложно убедиться, что построенный ДКА принимает $A \cap B$. Таким образом, мы построили два разных автомата, принимающих один и тот же язык.

Задача 1.21. **(a)** Модернизируйте конструкцию из доказательства 1.1, чтобы построить ДКА, принимающий $A \cup B$.

(b) Если $A \subset \Sigma^*$ регулярен, то и $\overline{A}$ является регулярным.

(c) Если языки A, B регулярны, то регулярен и $A \setminus B$.

Задача 1.22. Постройте для следующих пар языков A и B детерминированные конечные автоматы, принимающие $A \cap B$ и $A \cup B$:

(a) $A = \{\omega \in \{a,b\}^* | \#_a(\omega) + 2\#_b(\omega) = 0 \bmod 5\}$, $B = \{\omega \in \{a,b\}^* | \#_a(\omega) \geqslant 2,\ \#_b(\omega) \geqslant 2\}$;

(b) $A = \{abw | w \in \{a,b\}^*\}$, $B = \{w \in \{a,b\}^* | w$ содержит подслово $aba\}$.

Задача 1.23. Пусть A — ДКА, имеющий n состояний, а $L(A)$ содержит хотя бы один *палиндром*, то есть слово, равное своему обращению: $w = w^R$. Покажите, что $L(A)$ содержит хотя бы один палиндром длины не более $2n^2$.

Матрицы инцидентности

Равно как и ориентированные графы, конечные автоматы можно задавать с помощью матриц инцидентности. Пусть конечный автомат $Aut = (Q, \Sigma, Start, Final, T)$ (не обязательно детерминированный) имеет N состояний, тогда функцию перехода T можно задать матрицей инцидентности диаграммы Мура. А именно, пусть $M \in Mat_N(2^{\Sigma^*})$ — матрица, заданная

$$\forall q_1, q_2 \in Q \qquad M_{q_1,q_2} = \{x \in \Sigma | q_2 \in T(q_1, x)\}$$

Например, автомат выше ПИКЧА задается матрицей

$$\begin{bmatrix} \{a\} & \{b\} & \varnothing & \varnothing \\ \varnothing & \varnothing & \{a\} & \{b\} \\ \{a\} & \{b\} & \varnothing & \varnothing \\ \varnothing & \varnothing & \varnothing & \{a,b\} \end{bmatrix}$$

Теперь обобщим стандартные матричные операции сложения и умножения:

$$(A+B)_{q_1,q_2} = A_{q_1,q_2} \cup B_{q_1,q_2}; \qquad (AB)_{q_1,q_2} = \bigcup_{q \in Q} A_{q_1,q} B_{q,q_2}$$

Единичную матрицу введем, как и полагается, так:

$$\mathrm{Id}_{q_1,q_2} = \begin{cases} \{\epsilon\} & , q_1 = q_2 \\ \varnothing & q_1 \neq q_2 \end{cases}$$

А степень определяется индуктивно $A^0 = \mathrm{Id}; \qquad A^{n+1} = A^n A.$

Задача 1.24. Докажите следующие утверждения:

(a) $(A^n)_{q_1,q_2} = \{w \in \Sigma^* | |w| = n, q_2 \in \widetilde{T}(q_1, w)\}$;

(b) $L(Aut) = \bigcup\limits_{s \in Start} \bigcup\limits_{f \in Final} \bigcup\limits_{n \in \mathbb{N}} (A^n)_{s,f}$

Задача 1.25. Пусть $L \subset \Sigma^*$ регулярен. Покажите, что следующие языки регулярны:

(a) $\{x \in \Sigma^* | \exists y \in \Sigma^* : |y| = 2^{|x|},\ xy \in L\}$;

(b) $\{x \in \Sigma^* | \exists y \in \Sigma^* : |y| = p(|x|),\ y \in L\}$, где $p : \mathbb{N} \to \mathbb{N}$ — многочлен.

Можно рассматривать также матрицы с коэффициентами в $\mathbb{Z}\langle\langle\Sigma\rangle\rangle$, то есть $M_{q_1,q_2} = \sum_{x \in \Sigma,\ T(q_1,x)=q_2} x$. Для автомата с картинки имеем

$$\begin{bmatrix} a & b & 0 & 0 \\ 0 & 0 & a & b \\ a & b & 0 & 0 \\ 0 & 0 & 0 & a+b \end{bmatrix}$$

Для таких матриц можно переопределить хорошо известные матричные функции. След матрицы можно определить обычным образом $\mathrm{tr}(M) = \sum_i M_{ii}$. ...ДЕТЕРМИНАНТ...

Задача 1.26. Докажите, что $\det(\exp(M)) = \exp(\mathrm{tr}(M))$ для матриц конечных автоматов.

Глава 2

Недетерминированные конечные автоматы

Теперь мы рассмотрим недетрминированные конечные автоматы. Они отличаются от детерминированных конечных автоматов тем, что из некоторых состояний по некоторым символам можно совершить переходы в разные состояния одновременно. Мы будем считать, что недетерминированный автомат «находится одновременно» в разных состояниях.

Определение. *Недетерминированный конечный автомат* — $(Q, \Sigma, Start, Final, T)$, где

- Q — конечное множество состояний,
- Σ — алфавит, конечное множество символов,
- $Start \subseteq Q$ — подмножество стартовых состояний,
- $Final \subseteq Q$ — подмножество финальных состояний,
- T — таблица переходов, отображение $Q \times \Sigma \to 2^Q$:

$$T(s_i, \lambda) = \{s_j\} \tag{2.1}$$

Конфигурация ДКА есть подмножество $Q \times \Sigma^*$. Таблица переходов T задет отображение пространства конфигураций:

$$\forall q \in Q, x \in \Sigma, w \in \Sigma^* \qquad (q, xw) \mapsto \{(s, w) | s \in T(q, x)\}$$

Тогда индуктивно продолжим таблицу переходов T до отображения $\widetilde{T} : Q \times \Sigma^* \to Q$:

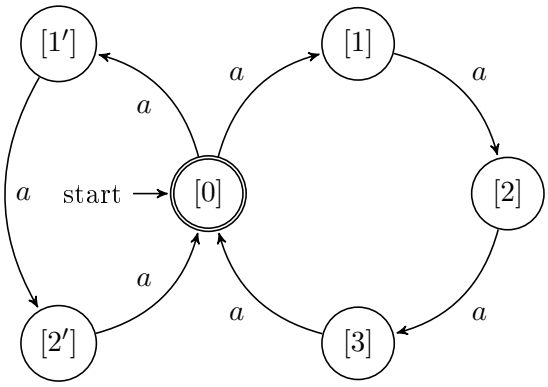
$$\forall q \in Q, x \in \Sigma, w \in \Sigma^* \qquad \widetilde{T}(q, \epsilon) = q,\ \widetilde{T}(q, xw) = \{T(s, x) | s \in \widetilde{T}(q, w)\} \tag{2.2}$$

Как можно заметить, $\widetilde{T}(q, w) \subset Q$ есть множество состояний, в которые можно попасть из q по слову w. Это можно проверить индукцией по $|w|$, доказательство предоставляется читателю.

Слово $\omega \in \Sigma^*$ *принимается* автоматом A, если $\widetilde{T}(Start, w) \cap Final \neq \varnothing$, то есть при переходах по символам этого слова по таблице переходов $T(A)$ автомат A остановится в множестве состояний, содержащем хотя бы одно финальное состояние. *Язык автомата* $L(A)$ — множество всех слов, принимаемых автоматом.

Изображать недетерминированные автоматы мы также будем диаграммами Мура. Слово принимается НКА, если существует хотя бы один путь в этой диаграмме из некоторого стартового состояния $s \in Start$ в некоторое финальное $f \in Final$, побуквенно помеченный этим словом. Возможно, что таких путей несколько, и они ведут из разных стартовых в разные финальные состояния.

Пример 11. Рассмотрим следующий недетерминированный автомат A:



Формально говоря, $A = (Q, \Sigma, Start, Final, T)$, где

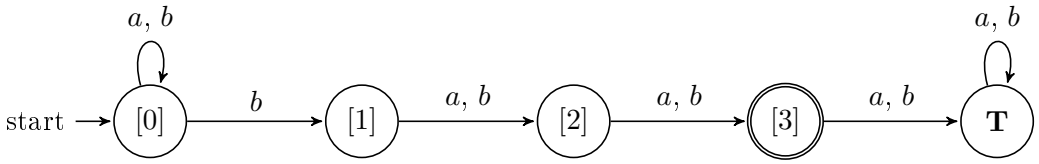
- $Q = \{[0], [1], [2], [3], [1'], [2']\}$;
- $\Sigma = \{a\}$;
- $Start = \{[0]\}$;
- $Final = \{[0]\}$;
- T — таблица переходов вида

$$[0] \xrightarrow{a} [1],\ [1] \xrightarrow{a} [2],\ [2] \xrightarrow{a} [3],\ [3] \xrightarrow{a} [0] \tag{2.3}$$

$$[0] \xrightarrow{a} [1'],\ [1'] \xrightarrow{a} [2'],\ [2'] \xrightarrow{a} [0] \tag{2.4}$$

Найдем язык $L(A)$ этого автомата. Слово a^x принимается НКА A титтк на диаграмме Мура существует путь длины N, начинающийся и заканчивающийся в состоянии [0]. Этот путь должен проходить по циклам $[0] \to [1] \to [2] \to [3]$ и $[0] \to [1'] \to [2']$ и содержать каждый из них целое число раз (возможно, нулевое): этот путь можно разбить на интервалы, начинающиеся и заканчивающиеся в [0] и не содержащие больше [0]. Таким образом, $N = 3m + 4n$ для некоторых $m, n \in \mathbb{Z}_{\geqslant 0}$, то есть $L(A) = \{a^{3m+4n} | m, n \in \mathbb{Z}_{\geqslant 0}\}$.

Пример 12. Язык всех слов из букв $\{a, b\}$, у которых b стоит на третьем с конца месте, распознается некоторым недетерминированным автоматом:



Формально говоря, $A = (Q, \Sigma, Start, Final, T)$, где

- $Q = \{[i] | i \in \{0, 1, 2, 3\} \cup \{\mathbf{T}\}$;
- $\Sigma = \{a, b\}$;
- $Start = \{[0]\}$;
- $Final = \{[3]\}$;
- T — таблица переходов вида

$$\forall x \in \Sigma \qquad [0] \xrightarrow{b} [1];\ [0] \xrightarrow{x} [0],\ [1] \xrightarrow{x} [2],\ [2] \xrightarrow{x} [3],\ [3] \xrightarrow{x} \mathbf{T},\ \mathbf{T} \xrightarrow{x} \mathbf{T} \tag{2.5}$$

Докажем, что этот автомат принимает только слова вида wbx_1x_2, где $x_1,\ x_2 \in \Sigma$, $w \in \Sigma^*$. Действительно, рассмотрим $\tilde{u}$ — суффикс u длины 3. Если $\tilde{u} = bx_1x_2$, то u принимается автоматом: очевидно, $[0] \in \widetilde{T}(Start, \omega)$ для любого слова ω, а так как $[1] \in T([0], b)$, то $[3] \in \widetilde{T}([0], bx_1x_2)$. Если же $\tilde{u} = ax_1x_2$, то u не принимается автоматом: последние три буквы образуют слово ax_1x_2, тогда $\widetilde{T}([0], ax_1x_2) = \widetilde{T}([0], x_1x_2) = [0]$ и $\widetilde{T}([i], ax_1x_2) = [3]$ для любого $i \neq 0$, ни одно из этих трех состояний не является финальным.

Пример 13. Пусть язык $L \subset \{a, b\}^*$ распознается некоторым детерминированным конечным автоматом A. Язык $\sqrt{L} = \{w \in \Sigma^* | w^2 \in L\}$ распознается некоторым недетерминированным автоматом. Идея заключается в том, чтобы угадать состояние, в котором автомат A пройдет середину слова $ww \in L$; это будет некоторое состояние q такое, что $Start \xrightarrow{w} q \xrightarrow{w} f \in Final$, a priori оно может быть любым.

Опишем конструкцию явно. ПИКЧА! Пусть $A = (Q_A, \Sigma, Start_A, Final_A, T_A)$, тогда построим НКА $\widehat{A}$:

- $Q = Q_A^3 = Q_A \times Q_A \times Q_A$;
- $\Sigma = \{a, b\}$;
- $Start = \{Start_A\} \times \{(q, q) | q \in Q_A\}$;
- $Final = \{(q, q) | q \in Q_A\} \times Final_A$;
- T — таблица переходов вида

$$\forall q, \in Q, x \in \Sigma\ (q_1, q, q_2) \xrightarrow{x} (T(q_1, x), q, T(q_2, x)) \tag{2.6}$$

Недетерминированность этого автомата — в недетерминированности старта: мы пытаемся угадать состояние, в котором A оказался, прочитав середину слова w^2, которое он примет в итоге. Теперь докажем, что такая конструкция действительно принимает лишь $\sqrt{L}$. Слово w принимается $\hat{A}$ титтк $\exists q \in Q_A$ такое, что $\widetilde{T}_A(Start, w) = q$ и $\widetilde{T}_A(q, w) \in Final$. Такое q существует титтк слово w^2 принимается автоматом A и $\widetilde{T}_A(Start, w) = q$.

Теперь мы докажем, что языки, принимаемые НКА, регулярны.

Теорема 2.1. *Для любого НКА A_n есть ДКА A_d такой, что $L(A_n) = L(A_d)$.*

Доказательство. Построим по A_n детерминированный A_d. Для этого сделаем простое наблюдение: в НКА однозначно осуществляется переход по букве из одного состояния в подмножество состояний, поэтому для любого $S \subset Q(A_d)$ образ при переходе по букве определен однозначно:

$$\hat{T}(S, a) = \bigcup_{s_i \in S} T(s_i, a),\ \forall a \in \Sigma, S \subset Q(A) \tag{2.7}$$

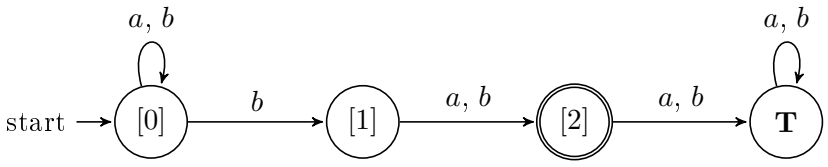
Это отображение в нашем случае продолжается до

$$\widetilde{T}_d(S, ax) = \bigcup_{s_i \in T(S,x)} T(s_i, x),\ \forall a \in \Sigma,\ x \in \Sigma^*,\ S \subset Q(A) \tag{2.8}$$

Иными словами, мы переходим одновременно во все состояния, в которые можем перейти в заданном НКА переходом по данной букве. Состояния A_d, таким образом, будут подмножествами состояний A_n, финальными мы объявим подмножества $S_{Final} \subset Q(A_n)$, содержащие финальные состояния. Так что если $A_n = (Q, \Sigma, Start, Final, T)$, то $A_d = (2^Q, \Sigma, \{Start\}, S_{Final}, \hat{T})$. Языки этих автоматов совпадают: $w \in L(A_n)$ титтк $\widetilde{T}_d(\{Start\}, w) \in S_{Final}$, что верно титтк существует подмножество $\widetilde{T}_d(\{Start\}, w) \subset Q$, которое есть $\widetilde{T}(Start, w)$ для НКА A_n и содержит хотя бы одно финальное состояние, то есть $w \in L(A_n)$. □

Так как детерминированный конечный автомат является частным случаем НКА, то из теоремы следует, что ДКА и НКА распознают один и тот же класс языков, то есть регулярные. Если НКА A_n имеет k состояний, то ДКА A_d, построенный по нему, будет иметь не более 2^k состояний. Впоследствии мы докажем, что эта оценка асимптотически точна, то есть существует серия языков $L_k \subset \{a,b\}^*$ для $k \in \mathbb{N}$ такая, что существует НКА, принимающий L_k и имеющий $O(k)$ состояний, а любой ДКА, принимающий его, имеет $\Theta(2^k)$ состояний.

Пример 14. Рассмотрим НКА, принимающий язык всех слов из букв $\{a,b\}$, у которых b стоит на предпоследнем месте:



Теперь построим детерминированный автомат, принимающий тот же язык. ПИКЧА!

Задача 2.1. Постройте НКА, принимающий $L_k = \{w \in \{a,b\}^* | w = u_1 b u_2, |u_2| = k-1\}$, и детерминизируйте его.

Полученный НКА имеет $k+1$ состояние, а $\Omega(2^k)$ состояний. Доказывать, что любой ДКА, принимающий L_k, имеет столько состояний, мы научимся чуть позже.

Задача 2.2. Покажите, что приведенный выше алгоритм детерминизации совершает $O(|\Sigma| N 2^N)$ операций, где N — число состояний исходного автомата.

Задача 2.3. **(a)** Покажите, что любой конечный язык $\{w_1, \dots w_n\} \subset \Sigma^*$ регулярен.

(b) Постройте НКА, принимающий $\{ba^3b, a^2b^2, aba^2\}$, и детерминизируйте его.

Задача 2.4. Пусть $L \subset \Sigma^*$ регулярен. Покажите, что L^R также регулярен.

Задача 2.5. Вспомним, что в задаче 0.6 мы ввели отношение квазипорядка на Σ^*: $x \preceq y$, если x получается из y удалением нескольких символов.

(a) Пусть $L_w = \{u \in \Sigma^* \mid w \preceq u\}$ Покажите его регулярность.

(b) Скажем, что язык L $\preceq$-*замкнут вверх*, если вместе с любым словом из L содержатся все слова, большие его: $x \in L$, $x \preceq y \Rightarrow y \in L$. Докажите, что любой $\preceq$-замкнутый вверх язык регулярен.

Задача 2.6. Пусть язык $L \subset \Sigma^*$ регулярен. Докажите, что и $FH(L) = \{x \in \Sigma^* | \exists y \in \Sigma^* \ |x| = |y|, \ xy \in L\}$, язык первых половинок слов L, является регулярным.

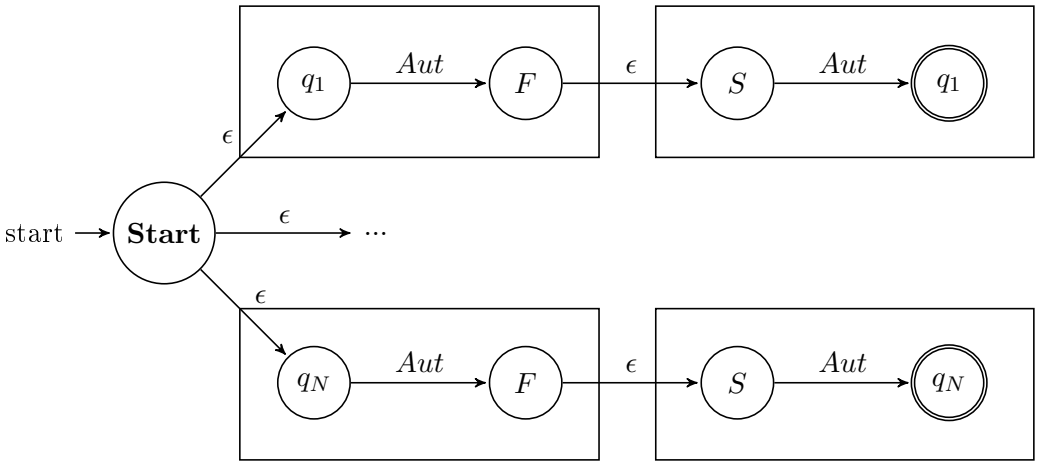
Задача 2.7. Пусть язык $L \subset \Sigma^*$ регулярен. Докажите, что и $MT(L) = \{y \in \Sigma^* | \exists x,z \in \Sigma^* \ |x| = |y| = |z|, \ xyz \in L\}$, язык серединок L, является регулярным.

Задача 2.8. Для любого $L \subset \Sigma^*$ определим *циклическое замыкание* $\Delta(L)$ как наименьший язык, содержащий L и удовлетворяющий условию

$$\forall x,y \in \Sigma^+, \ xy \in \Delta(L) \iff yx \in \Delta(L) \tag{2.9}$$

Пусть L регулярный. Верно ли, что $\Delta(L)$ регулярный?

Решение. Верно. Существует ДКА Aut, принимающий L. Мы будем пользоваться его минимальностью, полагая, что все состояния достижимы из старта. Построим НКА Aut_Δ, принимающий $\Delta(L)$:



Если слово регулярного языка допускает разбиение xy, значит, автомат Aut находился в каком-то состоянии, прочитав x, а Aut_Δ пытается его недетерминированно угадать, начиная читать y с какого-то состояния, придя в финал и затем начиная читать префикс. Если $xy \in L$, то Aut_Δ должен закончить в том же состоянии, что и Aut, прочитавший x; Aut_Δ должен просто помнить, какое состояние было гипотетически стартовым для чтения суффикса и был ли прочитан суффикс.

Формально объясняем так. Пусть Aut, ДКА, принимающий L, задан следующим образом:

- $Q = \{q_1, \dots q_N\}$ — множество состояний Aut,
- Σ — оригинальный алфавит,
- $Start \in Q$ — стартовое состояние,
- $Final \subset Q$ — подмножество финальных состояний,
- T - таблица переходов Aut,

Опеределим Aut_Δ таким образом:

- $Q_\Delta = \{\mathbf{Start}\} \cup (Q \times [1; N] \times \{0; 1\})$,
- Σ - алфавит, данный в условии,
- $Start_\Delta = \mathbf{Start}$ — выделяем отдельно стартовое состояние,
- $Final_\Delta = \{(q_i, i, 1) | i \in [1; N]\}$,
- T_Δ есть объединение старых переходов

$$\forall x \in \Sigma, j \in [1; N], k \in \{0, 1\}\ ((q_i, j, k), x) \to (T(q_i, x), j, k)$$

и новых переходов по ϵ

$$\forall f \in Final, j \in [1; N]\ ((f, j, 0), \epsilon) \to (Start, j, 1);\ (\mathbf{Start}, \epsilon) \to \{(q_i, i, 0) | i \in [1; N]\}$$

Проверим корректность нашей конструкции. Автомат Aut_Δ, работая на слове w, остановится в множестве состояний, содержащем некоторый $(q_i, i, 1)$, титтк существуют $x, y \in \Sigma^*$ такие, что $w = xy$ и автомат Aut,

- читая x и начиная работу в q_i, закончит в некотором $F \in Final$;
- читая y и начиная работу в $Start$, закончит в q_i.

Это верно, так как Aut_Δ может находиться в состояниях $(q_i, j, 1)$, читая слово w, титтк некоторый префикс является суффиксом некоторого слова регулярного языка: в такое состояние Aut_Δ мог попасть, только проходя перед этим $(Start, j, 1)$, а значит, и $(F, j, 0)$ для некоторого $F \in Final$, то есть титтк слово, прочитанное по пути из $(q_j, j, 0)$ в $(F, j, 0)$, есть суффикс некоторого слова из L (В этом моменте мы существенно пользуемся тем, что состояние q_j достижимо из стартового состояния Aut.) Соответственно, если x является некоторым суффиксом некоторого слова $w \in L$, то y, прочитанный по пути из $(Start, j, 1)$ в $(q_j, j, 1)$, можно считать префиксом некоторого слова из языка L. Вместе yx образуют слово, по которому ДКА Aut, начиная в $Start$, проходит в q_j, а затем и в некоторое финальное состояние. Если же $x, y \in \Sigma^*$, заданные условием выше, существуют, то Aut_Δ очевидно примет слово yx, а $xy \in L$. □

Задача 2.9. Дан $L \subset \Sigma^*$. Определим

$$\sqrt[n]{L} = \{w \in \Sigma^* | w^n \in L\}.$$

Покажите, что если L регулярен, то

(a) $\sqrt[n]{L}$ регулярен для любого $n \geqslant 1$;

(b) $\bigcup\limits_{n \geqslant 1} \sqrt[n]{L} = \bigcup\limits_{n \leqslant |Q|} \sqrt[n]{L}$.

регулярны.

Задача 2.10 (Seiferas, McNaughton [?seifmcn])**.** Скажем, что функция $f : \mathbb{Z}_{\geqslant 0} \to \mathbb{Z}_{\geqslant 0}$ *сохраняет регулярность*, если для любого регулярного $L \subset \Sigma^*$ язык

$$\{x \mid \exists y,\ |y| = f(|x|),\ xy \in L\}$$

. Скажем, что функция $f : \mathbb{Z}_{\geqslant 0} \to \mathbb{Z}_{\geqslant 0}$ *слабо сохраняет регулярность*, если для любого регулярного $L \subset \Sigma^*$ язык

$$\{x \mid \exists y,\ |y| = f(|x|),\ y \in L\}$$

. Покажите, что следующие утверждения эквивалентны:

(a) f сохраняет регулярность;

(b) f слабо сохраняет регулярность;

(c) для любого асимптотически периодического множества S множество $f^{-1}(S)$ также асимптотически периодическое;

(d) для любого $n \in \mathbb{Z}_{\geqslant 0}$ множество $f^{-1}(n)$ асимптотически периодическое, а для любого $M \in \mathbb{N}$ имеем

$$\exists q \in \mathbb{N},\ \exists m \in \mathbb{N}\ \forall n \geqslant m \quad f(n) = f(n+q) \bmod p$$

Задача 2.11. Определим *альтернированный конечный автомат* как пятерку

$$AltAut = (Q, \Sigma, T, F_{Start}, F_{Final}),$$

где

- Q — конечное множество состояний,
- Σ — алфавит,
- T — таблица переходов, отображение $\Sigma \to ((Q \to \{0,1\}) \to (Q \to \{0,1\}))$:

$$T(x) : f(q, x) \mapsto T[f(q, x)],\ f \in \{0,1\}^Q \tag{2.10}$$

- $F_{Final} : 2^{Q \to \{0,1\}}$ — *условие принятия*, множество «финальных» распределений,
- $F_{Start} : Q \to \{0,1\}$ — характеристическая функция финальных состояний.

Здесь состояния суть распределение нулей и единиц на конечном множестве. Каждый переход по букве x меняет распределение указанным выше образом. Начальное распределение есть F_{Start}, финальное — F_{Final}. Как и в классической ситуации, индуктивно определяется функция $\widetilde{T} : (Q \times \Sigma^*) \to ((Q \to \{0,1\}) \to \{0,1\})$:

$$\widetilde{T}(\epsilon) : F_{Start} \mapsto F_{Start},\ \widetilde{T}(xw) : F \mapsto T(x)(F) \tag{2.11}$$

Сокращенно будем писать $F \xrightarrow{w} G$, если $\widetilde{T}$ Говорим, что слово $w \in \Sigma^*$ принимается автоматом $AltAut$, если $\widetilde{T}(F_{Start}) \in F_{Final}$. Покажите, что язык L принимается АКА с N состояниями титтк A^R принимается некоторым ДКА с 2^N состояниями.

Задача 2.12 (Matos [?matos]). Для слова $w = w[1] \dots w[n]$ определим

$$e_r^q(w) = \{w[r]w[k+r] \dots w[k + r\frac{n-r}{q}] \mid \exists j \in \mathbb{Z}_{\geqslant 0}, n = kj + r\}, \quad q, r \in \mathbb{Z}_{\geqslant 0}$$

и

$$pad_a^q(w) = wa^r,\ r = \min_{\mathbb{Z}_{\geqslant 0}}\{x \mid x + |w| \vdots q\}.$$

Например,

$$e_1^2(\underline{a}b\underline{b}a\underline{a}b) = aba, \quad pad_a^5(abbaab) = abbaab\underline{aaaa}.$$

Для языка $L \subset \Sigma^*$ введем $e_r^q(L) = \{e_r^q(w) \mid w \in L\}$ и $pad^q(L) = \{pad_a^q(w) \mid w \in L,\ a \in \Sigma\}$. Пусть L регулярен, покажите регулярность

(a) $e_2^1(L)$ и $e_2^2(L)$;

(b) $e_r^q(L)$;

(c) $pad^q(L)$.

Задача 2.13. Определим *шаффл* двух слов индуктивно:

$$\forall x, y \in \Sigma^* \qquad x||\epsilon = \{x\},\ \epsilon||y = \{y\},\ (xa)||(yb) = (x||yb)a + (xa||y)b$$

Иными словами, шаффл x и y — это множество слов, полученное «вставкой» этих слов друг в друга. Покажите, что если A и B регулярны, то

$$A||B = \bigcup_{x \in A, y \in B} x||y \tag{2.12}$$

также регулярен.

Задача 2.14. Обобщим конструкцию из задачи ... Пусть $\Delta = \{1, 2, \dots k\} \subset \mathbb{N}$. Для $w \in \Sigma^*$ и $u \in \Delta^*$ одинаковой длины n определим $w^u = w[1]^{u[1]} \dots w[n]^{u[n]}$. Например, $bab^{124} = ba^2b^4$. Соответственно, для $A \subset \Sigma^*$ и $B \subset \Delta^*$ определим $A^B = \{w^u \mid w \in A,\ u \in B\}$. Покажите, что если A и B регулярны, то и A^B регулярен.

Задача 2.15. Дан $\Sigma = \{a, b\}$. Определим на словах одинаковой длины *метрику Хэмминга* $H(w_1, w_2)$ как число позиций, в которых эти слова различаются. Если $|w_1| \neq |w_2|$, то зададим $H(w_1, w_2) = \infty$. Для языка $A \in \Sigma^*$ определим

$$H(w, A) = \min_{u \in A} H(w, u)$$

Покажите, что для любого регулярного L язык $Ham(L, k) = \{w \in \Sigma^* | H(w, L) \leqslant k\}$ регулярен.

Решение. Пусть $Aut = (Q, \Sigma, Start, Final, T)$ — полный ДКА, принимающий L; построим НКА, прнимающий $Ham(L, k)$. Идеология простая: воспользоваться силой декаротова произведения, чтобы помнить и состояние в Aut, и число ошибок в слове. Формально говоря, автомат $\widehat{Aut} = (\widehat{Q}, \Sigma, \widehat{Start}, \widehat{Final}, \widehat{T})$, где

- $\widehat{Q} = (Q \times [0; k]) \cup \{\mathbf{TRASH}\}$, где $[0; k] \subset \mathbb{Z}$;
- Σ — старый добрый алфавит;
- $\widehat{Start} = (Start, 0)$ — начальное состояние;
- $\widehat{Final} = \{S \subset \widehat{Q} | \exists f \in Final, j \in [0; k]\ (f, j) \in S\}$ — остановимся на слове w, если есть слово не далее чем на j, лежащее в языке L;
- $\widehat{T}$ — таблица переходов вида

$$((s, i), a) \to \begin{cases} \{(T(s, a), i), (T(s, b), i + 1)\}, & i < k \\ \{(T(s, a), i), \mathbf{TRASH}\}, & i = k \end{cases},$$

$$((s, i), b) \to \begin{cases} \{(T(s, b), i), (T(s, a), i + 1)\}, & i < k \\ \{(T(s, b), i), \mathbf{TRASH}\}, & i = k \end{cases},$$

$$\forall x \in \Sigma\ (\mathbf{TRASH}, x) \to \mathbf{TRASH}$$

ведь если $H(\alpha, \beta) = i$, то $H(\alpha, \beta) \in \{i, i + 1\}$.

Теперь надо показать корректность такой конструкции. Действительно, автомат $\widehat{Aut}$, прочитав слово w длины n, остановится в множестве S_n таком, что

$$S_n \cap (Q \times [0; k]) = \{(q, i) | i \in [0; k],\ \exists u \in \Sigma^*\ |u| = n \text{ и } Aut \xrightarrow{u} q,\ H(w, u) = i\}$$

Докажем это по индукции. База при $n = 0$ очевидна; теперь докажем шаг индукции. Наше $w = \hat{w}x$ имеет длину $n + 1$, $x \in \Sigma$, а $\widehat{Aut}$, прочитав $\hat{w}$, остановится в состояниях (q, i) таких, что для некоторого $\hat{u} \in \Sigma^*$ длины n $Aut \xrightarrow{u} q$ и $H(w, u) = i$. Тогда $H(\hat{w}a, \hat{u}a) = H(\hat{w}b, \hat{u}b) = i$ и $H(\hat{w}a, \hat{u}a) = H(\hat{w}b, \hat{u}b) = i + 1$, и при прочтении буквы a имеем $((s, i), b) \to X$ такое, что $X \cap (Q \times [0; k]) = \{(T(s, a), i), (T(s, b), i + 1)\}$, здесь i и $i + 1$ есть в точности соответствующие расстояния Хэмминга; с буквой b получается аналогично.

Тогда $\widehat{Aut}$ остановится в некотором $S \subset \widehat{Q}$, содержащем некоторый (f, j) (где $\exists f \in Final, j \in [0; k]$) на слове w титтк $|w| = |u|$ для некоторого $u \in L$ и $H(w, u) = j$. □

Задача 2.16. Пусть $R \subset \Sigma^*$ — регулярный. Верно ли, что $\{w \in \Sigma^* | w^{|w|} \in R\}$ — регулярный?

Конечность автомата

Конечность автомата важна. Покажем, что для любого языка можно построить бесконечный детерминированный автомат.

Определение. *Граф Кэли* группы G - граф (V, E), где V — элементы группы, а $E = (g_1, g_2)$, если $\exists w : g_1 = g_2 w^{\pm 1}$. Например, на картинке по центру изображен граф Кэли группы $\mathbb{Z}$.

Для моноидов граф Кэли определяется аналогично. Построим граф Кэли для Σ^*, объявив вершины, соответствующие словам языка, состояниями $Final$ и взяв e за состояние $Start$. Полученный граф и будет соответсвующим автоматом для языка.

Здесь же a posteriori выясняется, что число финальных состояний может быть бесконечным.

Задача 2.17. Распознаются ли этим автоматом $\{a^n b^n | n \in \mathbb{N}\}$? А $\{a^n b^n c^n | n \in \mathbb{N}\}$?

Задача 2.18. Любой ли язык распознается бесконечным детерминированным автоматом, если число финальных состояний конечно?

Задача 2.19. Существует ли язык, распознающийся бесконечным НКА и никаким бесконечным ДКА, число финальных состояний которого конечно?

ϵ-переходы и свойства замкнутости

Теперь же введем так называемые *ϵ-переходы*. Это переходы $q_1 \xrightarrow{\epsilon} q_2$ по пустой строке. Автомат с ϵ-переходами, попадая в состояние q, оказывается также во всех $s \in T(q, \epsilon)$. Этот аргумент позволяет понять, почему конечные автоматы с ϵ-переходами принимают те же языки, что и обычные конечные автоматы.

Лемма 2.1. *Для любого НКА с ϵ-переходами существует эквивалентный НКА, принимающий тот же язык. Таким образом, языки, распознаваемые ϵ-переходами, регулярны.*

Доказательство. По автомату $Aut_\epsilon = (Q, \Sigma, Start, Final, T)$ с ϵ-переходами можно построить эквивалентный НКА $Aut = (Q, \Sigma, Start, Final, \widehat{T})$, в котором к переходу $q_1 \xrightarrow{x} q_2$ добавляются $q_1 \xrightarrow{x} q$ для всех состояний q таких, что есть переход $q_2 \xrightarrow{\epsilon} q$. Формально,

$$\forall q \in Q, x \in \Sigma\ \widehat{T}(q, x) = T(q, x) \cup \{s \in Q | q \xrightarrow{\epsilon} s\}$$

Покажем индукцией по $|w|$, что $\forall S \subset Q$

$$Start \xrightarrow{w}_{Aut_\epsilon} S \iff Start \xrightarrow{w}_{Aut} S,$$

то есть что оба автомата одинаково работают на одном и том же слове w. На нулевом шаге оба автомата одновременно находятся в состоянии $Start$. Если же при прочтении слова w оба автомата окажутся в множестве состояний S, то при прочтении слова wx (где x — одна буква) они одновременно окажутся в

$$\bigcup_{q \in S} T(q, x) = \bigcup_{q \in S} T(q, x) \cup \{s \in Q | q \xrightarrow{\epsilon} s\}.$$

□

Хотя ϵ-переходы и не увеличивают вычислительной мощности, они оказываются регулярно полезны — как, например, при доказательстве следующего критически важного утверждения.

Утверждение 2.1. *Если A и B — регулярные языки, то $A + B$, AB и A^* также регулярны.*

Доказательство. Языки A и B регулярные, следовательно, принимаются некоторыми ДКА $\{Q_A, \Sigma, S_A, F_A, T_A\}$ и $\{Q_B, \Sigma, S_B, F_B, T_B\}$, мы построим по ним соответствующие НКА с ϵ-переходами.

(ПИКЧА)

Для $A \cup B$ мы построим автомат $Aut_{A \cup B} = \{Q_A \cup Q_B \cup \{*\}, \Sigma, \{*\}, F_A \cup F_B, T\}$, где

$$\forall q_A \in Q_A,\ q_B \in Q_B,\ x \in \Sigma\ T(*, \epsilon) = \{S_A, S_B\},\ T(q_A, x) = T_A(q_A, x),\ T(q_B, x) = T_B(q_B, x).$$

(ПИКЧА)

Для AB мы построим автомат $Aut_{AB}\{Q_A \cup Q_B, \Sigma, S_A, F_B, T\}$, где

$$\forall q_A \in Q_A,\ q_B \in Q_B,\ f \in F_A,\ x \in \Sigma\ T(q_A, x) = T_A(q_A, x),\ T(f, \epsilon) = S_B,\ T(q_B, x) = T_B(q_B, x).$$

(ПИКЧА)

Для A^* мы построим автомат $Aut_{A^*} = \{Q_A \cup \{*\}, \Sigma, \{*\}, \{*\}, T\}$, где

$$\forall q \in Q_A, x \in \Sigma\ T(q, x) = T_A(q, x),\ T(*, \epsilon) = S_A,\ \forall f \in F_A\ T(f, \epsilon) = *.$$

Для построенных автоматов с ϵ-переходами найдутся НКА, также принимающие $A + B$, AB и A^*, следовательно, $A + B$, AB и A^* окажутся регулярными. □

Задача 2.20. Покажите, что построенные НКА с ϵ-переходами действительно принимают именно $A + B$, AB и A^*.

ϵ-переходы в доказательстве выше нужны не просто так.

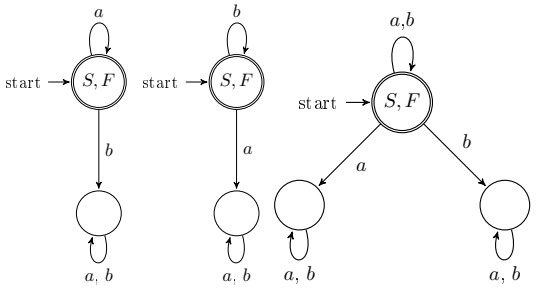
Задача 2.21. По двум ДКА $Aut_A = \{Q_A, \Sigma, S_A, F_A, T_A\}$ и $Aut_B = \{Q_B, \Sigma, S_B, F_B, T_B\}$, принимающим языки A и B, Вася строит НКА Aut_{A+B}, распознающий $A + B$.

(ПИКЧА) Построим НКА, принимающий $A + B$, склеив стартовые состояния автоматов, принимающих A и B. Формально говоря,

- $Q = \{Start\} \sqcup (Q_A \setminus S_A) \sqcup (Q_B \setminus S_B)$
- Σ — алфавит, совпадающий с алфавитом Aut_A и Aut_B
- $Start$ — новое стартовое состояние, «склейка» S_A и S_B
- $F = F_A \sqcup F_B$
- T — таблица переходов, все переходы которой описываются так:
  - $(p, x) \to q$, если есть переходы $(p, x) \to q$ в Aut_A или Aut_B, в которых $p, q \notin S_A$ или $p, q \notin S_B$;
  - $(Start, x) \to q$, если есть переходы $(S_A, x) \to q$ в Aut_A и $((S_B, x) \to q)$ в Aut_B.

Корректна ли его конструкция? Если да, то докажите, что $L(Aut_{A+B}) = A + B$, если нет, то приведите контрпример.

Решение. Ответ: нет.



(a) Aut_A (b) Aut_B (c) Aut_{A+B}

Ежу понятно, что ϵ-переходы использовались в классических доказательствах не просто так; в противном случае может существовать слово, прочитанное вдоль такого обхода по диаграмме Мура Aut_{A+B}: цикл внутри Aut_A из $Start$ в $Start$, проход по Aut_B в финальное состояние. Нужно подобрать так A и B, чтобы это слово не лежало в языке $A + B$. Пример двух автоматов на картинке.

На картинке выше $A = a^*$, $B = b^*$, а $L(Aut_{A+B}) = (a + b)^*$, конечно же, $(a + b)^* \neq a^* + b^*$: в $L(Aut_{A+B})$ есть слово ab, которого не может быть в $A + B$. □

Задача 2.22. По двум ДКА $Aut_A = \{Q_A, \Sigma, S_A, F_A, T_A\}$ и $Aut_B = \{Q_B, \Sigma, S_B, F_B, T_B\}$, принимающим языки A и B, Вася строит НКА Aut_{AB}, распознающий AB.

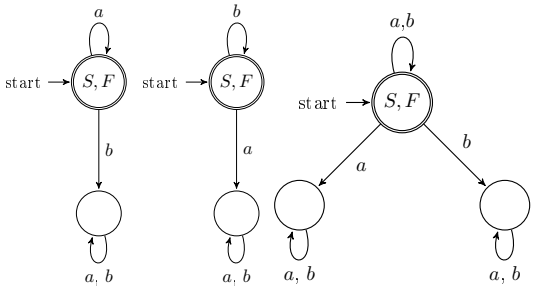
(ПИКЧА) Построим НКА, принимающий AB, склеив стартовое состояние Aut_B с финальными состояниями Aut_A. Формально говоря,

- $Q = \{q_{AB}\} \sqcup (Q_A \setminus F_A) \sqcup (Q_B \setminus S_B)$,
- Σ — алфавит, совпадающий с алфавитом Aut_A и Aut_B,
- $S = S_A$ — стартовое состояние Aut_A,
- $F = F_B$ — финальные состояния Aut_B,
- T — таблица переходов, все переходы которой описываются так:

– $(p, x) \to q$, если есть переходы $(p, x) \to q$ в Aut_A или Aut_B, в которых $p \notin F_A,\ q \notin S_B$;
– $(p, x) \to q_{AB}$, если есть переходы $((p, x) \to q)$ в Aut_A и $q \in F_A$;
– $(q_{AB}, x) \to q$, если есть переходы $((S_B, x) \to q)$ в Aut_B;
– $(q_{AB}, x) \to q_{AB}$, если есть переходы $((S_B, x) \to S_B)$ в Aut_B или $(p, x) \to q$ в Aut_A, где $p, q \in F_A$;

Корректна ли его конструкция? Если да, то докажите, что $L(Aut_{AB}) = AB$, если нет, то приведите контрпример.

Решение. Ответ: нет.



(a) Aut_A (b) Aut_B (c) Aut_{A+B}

Ежу понятно, что ϵ-переходы использовались в классических доказательствах не просто так; в противном случае может существовать слово, прочитанное вдоль такого обхода по диаграмме Мура Aut_{AB}: цикл внутри Aut_A из $Start$ в $Start$, проход по Aut_B в финальное состояние. Нужно подобрать так A и B, чтобы это слово не лежало в языке AB. Пример двух автоматов на картинке.

На картинке выше $A = a^*$, $B = b^*$, а $L(Aut_{AB}) = (a + b)^*$, конечно же, $(a + b)^* \neq a^*b^*$: в $L(Aut_{AB})$ есть слово ab, которого не может быть в $A + B$. □

Задача 2.23. Детерминизируйте автомат $Aut_{A \cup B}$ с ϵ-переходами, построенный в лемме 2.1. Сравните его с конструкцией из 1.1.

Задача 2.24. Пусть ДКА A и B имеют $N_A \geqslant 1$ и $N_B \geqslant 2$ состояний соответственно. Покажите, что существует ДКА, принимающий $L(A)L(B)$ и имеющий не более $N_A 2^{N_B} - 2^{N_B - 1}$ состояний.

Задача 2.25 (Yu, Zhuang, Salomaa [?yuzhuangsalomaa])**.** Пусть ДКА A имеет N_A состояний соответственно. Покажите, что существует ДКА, принимающий $L(A)^*$ и имеющий не более $2^{N_A - 1} + 2^{N_A - 2}$ состояний.

Мы завершим эту главу доказательством того, что регулярные языки замкнуты относительно морфизмов и прообразов морфизмов.

Теорема 2.2. *Пусть $h : \Sigma^* \to \Delta^*$ — морфизм. Если L регулярен, то и $h(L)$ регулярен.*

Доказательство. Язык L принимается некоторым ДКА Aut, построим недетерминированный конечный автомат $\widehat{Aut}$ такой, что $L(\widehat{Aut}) = h(L)$. ПИКЧА! Идея в том, чтобы вместо каждого перехода по букве $x \in \Sigma$ мы вклеиваем переход по буквам слова $h(x)$. Действительно, имея автомат $Aut = (Q, \Sigma, Start, Final, T)$, построим $\widehat{Aut} = (\widehat{Q}, \Delta, Start, Final, \widehat{T})$, где

- $\widehat{Q} = Q \cup \Big\{(q, T(q, x), i) \mid q \in Q, x \in \Sigma, i \in (0; |h(x)|)\Big\} \cup \{\mathbf{T}\}$ — к состояниям старого автомата добавили «промежуточные» состояния и «сток»,
- Δ — алфавит языка $h(L)$,
- $Start$ — стартовое состояние Aut,

- $Final$ — финальные состояния Aut,
- $\widehat{T}$ — таблица переходов, все переходы которой описываются так:
  - $\forall q \in Q \quad \widehat{T}(q,x) = \{(q, T(q,y), 1) \mid y \in \Sigma,\ x = h(y)[1]\} \cup \{T(q,y) \mid y \in \Sigma,\ |x| = 1,\ x = h(y)\}$;
  - $\widehat{T}((q, T(q,x), i), h(x)[i+1]) = (q, T(q,x), i+1)$;
  - $\forall q \in Q\ \widehat{T}(q, \epsilon) = q$, если $h(x) = \epsilon$ для некоторой $x \in \Sigma$;
  - все остальные переходы [по незайдествованным буквам] в $\mathbf{T}$.

Вообще этот автомат может быть недетерминированным: для $L = \{a,b\}^*$ и $h : \{a,b\}^* \to \{a\}^*$, $h(a) = a, h(b) = a^2$ получим следующие автоматы: ПИКЧА.

Докажем, что $L(\widehat{Aut}) = h(L)$. Действительно, если $w \in h(L)$, то $w = h(u) = h(u[1]) \dots h(u[|u|])$ для некоторого $u \in L$, тогда в диаграмме автомата $\widehat{Aut}$ существует путь

$$Start \xrightarrow{h(u[1])} \widetilde{T}(Start, u[1]) \xrightarrow{h(u[2])} \widetilde{T}(Start, u[1]u[2]) \dots \xrightarrow{h(u[|u|])} \widetilde{T}(Start, u[1] \dots u[|u|]) = \widetilde{T}(Start, u) \in Final.$$

Таким образом, $h(L) \subseteq L(\widehat{Aut})$. Теперь докажем, что $L(\widehat{Aut}) \subseteq h(L)$. Действительно, если слово w принимается $\widehat{Aut}$, то существует путь из стартового состояния в финальное, то есть последовательность состояний $S = \{s_0, \dots s_{|w|}\} \subset \widehat{Aut}$ такая, что $s_0 = Start$, $s_{i+1} \in \widehat{T}(s_i, w[i+1])$ для любого i и $s_{|w|} \in Final$. ПИКЧА! В этой последовательности нет состояния $\mathbf{T}$ — из него недостижимо ни одно состояние, кроме $\mathbf{T}$. Кроме того, эту последовательность можно разбить на интервалы между состояниями из Q, то есть существует $0 = i_0 \leqslant i_1 \leqslant \dots \leqslant i_k = |w|$ такая, что $\{s_{i_0}, \dots s_{i_k}\} \subset Q$, а остальные состояния — «промежуточные»:

$$S \setminus \{s_{i_0}, \dots s_{i_k}\} \subset \Big\{(q, T(q,x), i) \mid q \in Q, x \in \Sigma, i \in (0; |h(x)|)\Big\}.$$

Тогда вдоль любой подпоследовательности вида $\{s_{i_j}, s_{i_j+1}, \dots s_{i_{j+1}}\}$ читается подслово $u = h(x)$, $x \in \Sigma$: во-первых, первый переход имеет вид $s_{i_j} \to (s_{i_j}, s_{i_j+1}, 1)$ или $s_{i_j} \to s_{i_{j+1}}$, то есть первая буква подслова есть $h(x)[1]$, во-вторых, если $u[i] \neq h(x)[i]$, то следующее состояние в последовательности есть $\mathbf{T}$, которого не может быть. Таким образом, вдоль всего пути было прочитано слово $h(x_1) \dots h(x_k) = h(x_1 \dots x_k)$ для $x_1, \dots x_k \in \Sigma$, то есть слово из $h(L)$. □

Впоследствии мы получим еще одно доказательство этой теоремы.

Теорема 2.3. *Пусть $h : \Sigma^* \to \Delta^*$ — морфизм. Если L регулярен, то и $h^{-1}(L)$ регулярен.*

Доказательство. Язык L принимается некоторым ДКА Aut, построим недетерминированный конечный автомат $\widehat{Aut}$ такой, что $L(\widehat{Aut}) = h^{-1}(L)$. ПИКЧА! Мы перестроим Aut так, чтобы по букве x совершались переходы по слову $h(x)$. А именно, имея автомат $Aut = (Q, \Delta, Start, Final, T)$, построим $\widehat{Aut} = (Q, \Sigma, Start, Final, \widehat{T})$, где

- Q —состояния старого автомата,
- Σ — алфавит языка $h^{-1}(L)$,
- $Start$ — стартовое состояние Aut,
- $Final$ — финальные состояния Aut,
- $\widehat{T}$ — таблица переходов вида

$$\forall q \in Q, x \in \Sigma \quad \widehat{T}(q,x) = T(q, h(x))$$

Этот конечный автомат получается детерминированным. Докажем, что $L(\widehat{Aut}) = h^{-1}(L)$. Пусть $\widetilde{\widehat{T}}(q, w)$ — состояние, в котором $\widehat{Aut}$ окажется, перейдя из состояния q по буквам слова w; покажем индукцией по $|w|$, что $\widetilde{\widehat{T}}(q, w) = \widetilde{T}(q, h(w))$. Для слова длины 0 имеем $\widetilde{\widehat{T}}(q, \epsilon) = q = \widetilde{T}(q, \epsilon) = \widetilde{T}(q, h(\epsilon))$. Теперь в предположении $\forall w$, $|w| = n$ $\widetilde{\widehat{T}}(q, w) = \widetilde{T}(q, h(w))$ докажем переход. Слово длины $n + 1$ представим как wx, $|w| = n$, $x \in \Sigma$, поэтому

$$\widetilde{\widehat{T}}(q, wx) = \widehat{T}(\widetilde{\widehat{T}}(q, w), x) = \widehat{T}(\widetilde{T}(q, h(w)), x) = T(\widetilde{T}(q, h(w)), h(x)) = \widetilde{T}(q, h(w)h(x)) = \widetilde{T}(q, h(wx)).$$

Таким образом,

$$w \in L(\widehat{Aut}) \iff \widetilde{\widehat{T}}(q, w) \in Final \iff \widetilde{T}(q, h(w)) \in Final \iff h(w) \in L.$$

Значит, $w \in L(\widehat{Aut})$ титтк $w \in h^{-1}(L)$. □

Задача 2.26. Пусть язык L регулярен. Покажите, что множество длин слов L

$$\{l \mid \exists w \in L,\ |w| = l\} \subset \mathbb{Z}_{\geqslant 0}$$

является асимптотически периодическим.

Задача 2.27. Пусть язык L регулярен. Регулярен ли

$$\{w \in \Sigma^* | \exists u \in L :\ |w| - |u| = 1\}? \tag{2.13}$$

Задача 2.28. По аналогии с морфизмом определим *подстановку* $s : \Sigma^* \to \Sigma^*$ как мультипликативное отображение, заданное на алфавите следующим образом: $s(a)$ есть некоторый регулярный язык. Определим

$$s(L) = \{s(w) \mid w \in L\}, \quad s^{-1}(L) = \{w \mid s(w) \in L\}$$

Пусть L — регулярный язык, s — подстановка.

(a) Покажите, что $s(L)$ регулярен.

(b) Обязан ли $s^{-1}(L)$ быть регулярным?

Глава 3

Регулярные выражения и алгебра Клини

В этой главе мы рассмотрим иной способ задать регулярные языки. Матлогик Стивен Клини определил регулярные множества индуктивно и доказал, что они могут выражены через атомарные языки с помощью операций $+$ [объединение], $\cdot$ [конкатенация] и $*$ [звездочка Клини], полученное выражение называется регулярным. Впоследствии в той же статье Клини показал, что ...

Впоследствии регулярные выражения были использованы для создания текстовых редакторов ...

Определение. Множество *регулярных выражений* REG над алфавитом Σ — множество строчек над алфавитом $\Sigma \cup \{+, \cdot, *, (,)\}$, удовлетворяющее следующим правилам:

- $\varnothing, \epsilon, x$ для любой $x \in \Sigma$ — *атомарные* регулярные выражения;
- если α и β — регулярные выражения, то $\alpha + \beta, \alpha\beta, \alpha^*, (\alpha)$ — регулярные выражения;
- никаких других регулярных выражений нет, то есть любое регулярное выражение может быть получено из атомарных с помощью операций $+, \cdot, *$ и использования скобок.

Язык $L(\alpha)$ *регулярного выражения* α определяется реккурентно:

- $L(\varnothing) = \varnothing$, $L(\epsilon) = \{\epsilon\}$, $L(x) = \{x\}$ для любой $x \in \Sigma$;
- $L(\alpha + \beta) = L(\alpha) + L(\beta)$;
- $L(\alpha\beta) = L(\alpha)L(\beta)$;
- $L(\alpha^*) = (L(\alpha))^*$.

Язык $L \subset \Sigma^*$ называется *регулярным*, если он является языком некоторого регулярного выражения. Мы будем говорить, что $\alpha = \beta$, если их языки совпадают.

Под *длиной* $|\alpha|$ *регулярного выражения* α мы будем понимать длину α как слова над алфавитом $\Sigma \cup \{+, *, (,)\}$; знак умножения и ϵ вносят нулевой вклад в длину выражения.

В дальнейшем для краткости мы будем использовать выражение «язык α», имея в виду «язык $L(\alpha)$ регулярного выражения α», а под регулярным выражением Σ будем понимать сумму всех букв алфавита Σ.

В прошлых главах мы уже давали определение регулярности через конечные автоматы, чуть позже мы покажем эквивалентность всех данных нами определений регулярного языка. В нескольких следующих примерах мы покажем регулярность некоторых языков, предъявив для них регулярные выражения.

Пример 15. Рассмотрим регулярное выражение $(a+b)^*b^2a(ba+a^2b)^*$. Слово $a^2b^3a^3b^2aba$ лежит в языке данного регулярного выражения:

$$a^2b^3a^3b^2aba = \underbrace{a^2b}_{\in(a+b)^*} \cdot b^2a \cdot \underbrace{a^2b \cdot ba \cdot ba}_{\in(ba+a^2b)^*}$$

Пример 16. Язык всех слов, содержащих подслово $w \in \Sigma^*$, является регулярным: он может быть описан регулярным выражением $\Sigma^*w\Sigma^*$. Действительно, $u \in \Sigma^*$ содержит подслово w титтк для некоторого $k \in [1; |w|]$ имеет место $u[k] = w[1], \dots u[k+|w|-1] = w[|w|]$. Это эквивалентно $u = u_1wu_2$ для некоторых $u_1, u_2 \in \Sigma^*$ [возможно, пустых], то есть $w \in \Sigma^*w\Sigma^*$.

Пример 17. Докажем, что язык $L \subset \{a,b\}^*$ всех слов, содержащих три буквы b, является регулярным. Слово w, содержащее ровно три буквы b, должно иметь вид $a^xba^yba^zba^t$ для некоторых $x, y, z, t \in \mathbb{Z}_{\geqslant 0}$. Следовательно, $L = L(a^*ba^*ba^*ba^*)$.

Пример 18. Язык D_n всех слов из букв алфавита Σ, длина которых делится на $n \geqslant 1$, является языком регулярного выражения $(\Sigma^n)^*$: слово w имеет длину dn титтк $w = w_1 \dots w_d$, где $|w_1| = \dots = |w_d| = n$, то есть $w_1, \dots w_d \in \Sigma^n$. Следовательно, $|w| = dn$ титтк $w \in (\Sigma^n)^d$. Таким образом,

$$|w| \vdots n \iff \exists d \in \mathbb{N},\ |w| = dn \iff \exists d \in \mathbb{N},\ w \in (\Sigma^n)^d \iff w \in (\Sigma^n)^*$$

Пример 19. Пусть $w \in \Sigma^+$. Язык $Pref(w) = \{u \mid u \sqsubseteq w\}$ префиксов слова w конечен и таким образом является языком регулярного выражения

$$\epsilon + w[1] + \dots + w[1] \dots w[|w|]$$

В этом выражении $|w|$ знаков сложения, одно пустое слово и по одному слову каждой длины от 1 до $|w|$, значит, длина этого регулярного выражения равна $|w| + \sum\limits_{i=1}^{|w|} i = \binom{|w|+1}{2} - 1 = \Theta(|w|^2)$. Все буквы слова w, кроме последней, используются более одного раза; предъявим более короткое РВ, в котором каждая буква w встречается ровно один раз:

$$\beta(w) = \epsilon + w[1](\epsilon + w[2](\dots(\epsilon + w[|w|])\dots))$$

Например, для слова $w = a^2b^2ab$ имеем

$$\beta(a^2b^2ab) = \epsilon + a(\epsilon + a(\epsilon + b(\epsilon + b(\epsilon + a(\epsilon + b))))).$$

В $\beta(w)$ используется $|w|-1$ пара скобок, каждая буква w ровно по разу, и $|w|$ знаков сложения, таким образом, $|\beta(w)| = 4|w| - 2 = \Theta(|w|)$.

Задача 3.1. Докажите, что любое регулярное выражение α, описывающее $Pref(w)$, имеет длину $\Omega(|w|)$.

Решение. Так как язык $Pref(w)$ конечен, то α не содержит слагаемого вида $\gamma_1\beta^*\gamma_2$, где $L(\gamma_1), L(\gamma_2) \neq \varnothing$, а $L(\beta) \neq \varnothing$ или $L(\beta) \neq \epsilon$: в таком случае язык $L(\beta^*)$ бесконечен, тогда и язык $L(\gamma_1\beta^*\gamma_2)$ бесконечен. Это позволяет нам считать, что выражение α не содержит звездочку — в слагаемом вида $\gamma_1\beta^*\gamma_2$ либо одно из γ_i задает пустой язык [тогда слагаемое можно удалить], либо $\beta^* = \epsilon$ [и тогда можно сократить β^*]. В выражении α, не содержащем звездочек, должно быть хотя бы $|w|$ букв — иначе любое слово языка $L(\alpha)$ должно иметь длину, меньшую $|w|$, что не может правдой: $w \in Pref(w)$. □

Задача 3.2. Пусть $w \in \Sigma^+$. Постройте регулярные выражения, описывающие языки слов,

(a) имеющих префикс w;

(b) в которых есть подслово w, начинающееся с третьей позиции;

(c) содержащих хотя бы два непересекающихся вхождения w.

Задача 3.3. Определим $PreSuf(w)$ как слов, у которых существует и префикс, и суффикс, равный w. Постройте регулярные выражения для $PreSuf(w)$, где

(a) $w = a^3b^2$;

(b) $w = a^2b^3a^2$;

(c) $w = (ab)^4$.

Решение. 1. *Ответ:* $a^2b^3\Sigma^*a^2b^3 + a^2b^3$. Все слова вида $a^2b^3wa^2b^3$ лежат в $PreSuf(a^2b^3)$. Осталось показать, что любое слово из $PreSuf(a^2b^3)$ либо равно a^2b^3, либо имеет длину хотя бы 10. Слова длины 6 существовать не может ПИКЧА: у него четвертая буква равна и b, и a одновременно. Аналогично не бывает слов длины 7, 8 и 9 в $PreSuf(w)$.

2. *Ответ:* $a^2b^3a^2\Sigma^*a^2b^3a^2 + a^2b^3a^3b^3a^2 + a^2b^3a^2b^3a^2 + a^2b^3a^2$.

   Нужно быть осторожным: слово $a^2b^3a^2$ имеет нетривиальный перехлест ПИКЧА с самим собой. Поэтому аккуратно рассмотрим, какие слова могут иметь суффикс и префикс, равный $a^2b^3a^2$:

   - само слово $a^2b^3a^2$,
   - слово $a^2b^3a^2b^3a^2$, в котором искомые префикс и суффикс пересекаются по a^2,
   - слово $a^2b^3a^3b^3a^2$, в котором искомые префикс и суффикс пересекаются по a,
   - слово $a^2b^3a^2wa^2b^3a^2$ для любого $w \in \Sigma^*$, в котором искомые префикс и суффикс не пересекаются.

   Суммируем все это и получаем ответ.

3. *Ответ:* $(ab)^4\Sigma^*(ab)^4 + (ab)^4 + (ab)^5 + (ab)^6 + (ab)^7$. Получается рассуждениями, аналогичными предыдущему пункту.

□

Задача 3.4. Постройте регулярные выражения, описывающие следующие языки:

(a) на 2 месте от конца стоит b;

(b) на 3 месте от конца стоит b;

(c) на k месте от конца стоит b;

(d) есть суффикс вида abw, $w \in \Sigma^2$, то есть $|w| = 2$.

Задача 3.5. Постройте регулярные выражения, описывающие следующие языки:

(a) $\{\omega | \#_a(\omega) + 2\#_b(\omega) = 0 \bmod 5\}$;

(b) $\{\omega | 3\#_a(\omega) + 2\#_b(\omega) = 0 \bmod 6\}$;

(c) $\{\omega | A\#_a(\omega) + B\#_b(\omega) = k \bmod N\}$ для любых $A, B, k, N \in \mathbb{N}$.

Задача 3.6. $\Sigma = \{a, b, c\}$. Постройте регулярные выражения, описывающие следующие языки:

(a) язык слов, в которых сразу же после a идет b;

(b) язык слов, в которых сразу же после a не идет b.

Задача 3.7. Докажите, что любой бесконечный регулярный язык L содержит бесконечный регулярный подъязык R такой, что $L \setminus R$ бесконечный.

Задача 3.8. Пусть α – некоторое регулярное выражение. Укажите, как построить регулярное выражение языка $Pref(L(\alpha)) = \{x \in \Sigma^* \mid \exists y\, xy \in L(\alpha)\}$.

Задача 3.9. Пусть $w \in \Sigma^+$. Постройте регулярные выражение для языка $\Sigma^* \setminus \{w\}$, имеющее длину $O(|w|)$.

Задача 3.10. Проверить, что для любых регулярных выражений a, b, c имеют место равенства

- $a + (b + c) = (a + b) + c$, $a(bc) = (ab)c$;
- $a + b = b + a$;
- $a + a = a$;
- $a + \varnothing = a$;
- $a \cdot \epsilon = \epsilon \cdot a = a$, $a \cdot 0 = 0 \cdot a = 0$;
- $a(b + c) = ab + ac$, $(a + b)c = ac + bc$;
- $\epsilon + aa^* = \epsilon + a^*a = a^*$;
- $b + ac \subseteq c \iff a^*b \subseteq c$;
- $b + ca \subseteq c \iff ba^* \subseteq c$.

Задача 3.11. Скажем, что *дизъюнктивная нормальная форма* регулярного выражения α есть представление в виде $\alpha_1 + \ldots + \alpha_k$, где α_i не содержат сложения. Например, $(a^*b^*)^*$ — в дизъюнктивной нормальной форме, а $(a + b)^*$ — нет. Покажите, что любое РВ имеет эквивалентное ему выражение в ДНФ.

Мы уже давали в прошлой главе определение регулярных языков. Теперь мы покажем эквивалентность этих двух определений.

Теорема 3.1 (Kleene). *Следующие свойства языка L эквивалентны:*

- *L является языком некоторого регулярного выражения;*
- *L принимается некоторым конечным автоматом.*

Доказательство. Заметим, что атомные паттерны задают регулярные языки. Далее заметим, что если A, B регулярны, то и $AB, A + B, A^*$ регулярны.

Теперь же пусть язык L является языком некоторого детерминированного автомата $(Q, \Sigma, Start, Final, T)$ Построим по этому автомату регулярное выражение $L(A)$. Построим множество регулярных выражений $\{\alpha^X_{uv}\}$, где α^X_{uv} — регулярное выражение языка всех слов, которое получается прохождением пути вида

$$u \longrightarrow x_1 \longrightarrow x_2 \longrightarrow \cdots \longrightarrow v$$

где $u, v \in Q, x_1, \ldots x_k \in X \subset Q$. В самом деле, как мы ранее говорили, регулярный язык суть язык путей по конечному помеченному орграфу из одной вершины в другую; надо эти пути перечислить. Данные выражения строятся индуктивно по размеру X:

$$\alpha^{\varnothing}_{uv} = \begin{cases} \begin{cases} a_1 + \cdots + a_k \\ \varnothing \end{cases} & , a_i \text{ — символы, по которым есть переход из } u \text{ в } v,\ u \neq v \\ \begin{cases} a_1 + \cdots + a_k + \epsilon \\ \epsilon \end{cases} & , a_i \text{ — символы, по которым есть переход из } u \text{ в } v,\ u = v \end{cases}$$

Шаг индукции:

$$\alpha^X_{uv} = \alpha^{X-\{q\}}_{uv} + \alpha^{X-\{q\}}_{uq}(\alpha^{X-\{q\}}_{qq})^*\alpha^{X-\{q\}}_{qv} \qquad \forall q \in Q$$

В таком случае ответ

$$\alpha = \sum_{f \in Final} \alpha^Q_{Start,f}.$$

Докажем, что $L(\alpha)$ есть искомый язык. Для этого установим следующее: пусть L^X_{uv} — язык слов, читаемых в диаграмме Мура вдоль путей вида

$$u \longrightarrow x_1 \longrightarrow x_2 \longrightarrow \cdots \longrightarrow v$$

где $u, v \in Q, x_1, \ldots x_k \in X \subset Q$, формально,

$$L_{uv}^X = \{w \in \Sigma^* | u \xrightarrow{w} v,\ \forall p \supset w \exists x_i \in X\ u \xrightarrow{p} x_i\},$$

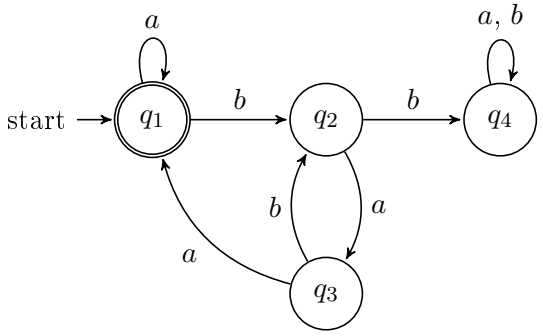
тогда $L_{uv}^X = L(\alpha_{uv}^X)$. Это доказывается индукцией по $|X|$. База при $|X| = 0$ очевидна. Пусть утверждение верно для некоторого X, добавим к нему некоторую вершину $s \in Q$. Любое слово, лежащее в $L_{uv}^{X \cup \{s\}}$, читается вдоль пути, который либо проходит через s, либо не проходит. В первом случае такие слова по предположению индукции образуют язык $\alpha_{us}^X (\alpha_{ss}^X)^* \alpha_{sv}^X$:

- путь, вдоль которого читаются такие слова, сначала посещает состояние s, соответствующие префиксы образуют α_{us}^X;
- такой путь может проходить через s сколь угодно раз, соответствующие подслова образуют $(\alpha_{ss}^X)^*$;
- последний раз посещая s, этот путь завершается в v, такие суффиксы образуют α_{sv}^X.

Слова же во втором случае образуют опять же по предположению α_{uv}^X. Таким образом, $L_{uv}^X = L(\alpha_{uv}^X)$, тогда $\alpha = \sum_{f \in Final} \alpha_{Start,f}^Q$. □

Чтобы вычислить регулярное выражение по автомату, необязательно считать все α_{uv}^X.

Пример:



Выберем q_2 за выбрасываемое состояние:

$$\alpha_{q_1,q_1}^{\{q_1,q_2,q_3\}} = \alpha_{q_1,q_1}^{\{q_1,q_3\}} + \alpha_{q_1,q_2}^{\{q_1,q_3\}} (\alpha_{q_2,q_2}^{\{q_1,q_3\}})^* \alpha_{q_2,q_1}^{\{q_1,q_3\}}$$

Можно и дальше продолжать разбиение, а можно заметить, что:

$\alpha_{q_1,q_1}^{\{q_1,q_3\}} = a^*$
$\alpha_{q_1,q_2}^{\{q_1,q_3\}} = a^*b$
$\alpha_{q_2,q_2}^{\{q_1,q_3\}} = \epsilon + ab + a^2a^*b$
$\alpha_{q_2,q_1}^{\{q_1,q_3\}} = a^2a^*$

Тогда имеем $a^* + a^*b(\epsilon + ab + a^2a^*b)^*a^2a^* = a^* + a^*b(aa^*b)^*a^2a^* = a^* + a^*ba(a^*ba)^*aa^* = \epsilon + (a + ba)^*a$.

Задача 3.12. Приведите альтернативное доказательство замкнутости регулярных языков относительно морфизмов.

Задача 3.13. Для каких $k, p, q \in \mathbb{Z}$ существует регулярный язык такой, что число слов длины не более n в нем равно

(a) $\Theta(n^k)$,

(b) $\Theta(2^{\frac{p}{q}n})$?

Задача 3.14. Пусть L — регулярный. Используя регулярные выражения, докажите, что язык

$$nL = \{\underbrace{a_1 \ldots a_1}_{n \text{ раз}} \underbrace{a_2 \ldots a_2}_{n \text{ раз}} \ldots \underbrace{a_k \ldots a_k}_{n \text{ раз}} | a_1 a_2 \ldots a_k \in L\}$$

является регулярным для любого $n \in \mathbb{N}$. Сравните с 1.11.

Задача 3.15. Рассмотрим $L = \{w \in \Sigma^* \mid \forall x \in \Sigma\ |w|_x = 1\}$. Покажите, что любое РВ, описывающее этот язык, имеет длину более $2^{|\Sigma|-1}$, в то время как существует РВ для $\Sigma^* \setminus L$ длины $O(|\Sigma|^2)$.

Задача 3.16. Для двух языков $A, B \subset \Sigma^*$ определим *частное* двух языков:

$$A/B = \{x \in \Sigma^* | \exists y \in B \qquad xy \in A\}$$

Покажите, что если A регулярный, то A/B регулярен для любого B.

Задача 3.17. Построим по ДКА $Aut = (Q, \Sigma, Start, Final, T)$ следующий орграф G: множеством вершин будет Q, а ребра суть

$$E = \{(q_i q_j) | q_i, q_j \in Q,\ \exists x \in \Sigma\ T(q_i, x) = q_j\}.$$

В графе G получилось 3 цикла. Верно ли, что $L(Aut)$ не может быть записан регулярным выражением, использующим не более одной звезды Клини?

Назовем $*$-*глубиной регулярного выражения* реккурентно определенную функцию $h : REG \to \mathbb{Z}$, удовлетворяющую следующим условиям:

- $h(\emptyset) = h(\epsilon) = h(x) = 0$ для $x \in \Sigma$,
- $h(\alpha + \beta) = h(\alpha\beta) = \max(h(\alpha), h(\beta))$,
- $h(\alpha^*) = h(\alpha) + 1$.

$*$-*глубиной регулярного языка* L назовем наименьшую $*$-глубину регулярного выражения, описывающего язык. Например, $(a^*)^* = a^*$, поэтому $*$-глубина языка $(a^*)^*$ равна 1.

Задача 3.18 (Eggan)**.** Теперь пусть имеется орграф G. Назовем *цикловым рангом* $r(G)$ функцию, заданную реккурентно:

- $r(G) = 0$, если граф не содержит циклов;
- для сильно связного G имеем $r(G) = 1 + \min_{v \in V} r(G - v)$, где вместе с вершиной v удаляются и все смежные ребра;
- если G не сильно связен, то $r(G) = \max r(G_i)$ для G_i — компонент сильной связности.

Докажите следующую *теорему Эггана*: $*$-глубина равна минимально возможному цикловому рангу недетерминированного конечного автомата с ϵ-переходами, принимающего данный язык.

Задача 3.19. Покажите, что $*$-глубина любого регулярного языка над $\{a\}$ конечна.

В то же время $*$-глубина регулярного языка над алфавитом большей мощности может быть сколь угодно большой. Эгган построил реккурентно заданное семейство регулярных языков $\{L_n\}$ такое, что $*$-глубина языка L_n равна n.

Задача 3.20 (Eggan, Salomaa [?salomaa])**. (a)** ...

Алгоритмы построения конечных автоматов по РВ

Теорема Клини допускает конструктивное доказательство: по регулярному выражению α можно явно построить конечный автомат, принимающий $L(\alpha)$. Существуют разные алгоритмы построения автомата по регулярному выражению; мы разберем некоторые, начав с алгоритма, предложенного Глушковым. Пусть дано регулярное выражение α над $\Sigma = \{a_1, \dots a_N\}$; по нему строим новый алфавит $\widehat{\Sigma} = \{a_{11}, \dots a_{1\#_{a_1}(\alpha)}, \dots, a_{N1}, \dots a_{N\#_{a_N}(\alpha)}\}$ и *линеаризованное* регулярное выражение $\hat{\alpha}$, полученное из α заменой букв a_i на a_{ir}, где r — номер вхождения буквы a_i в α. Например, для выражения $\alpha = ab(a^2 + bab)^*$ над $\Sigma = \{a, b\}$ получим $\hat{\alpha} = a_1 b_1 (a_2 a_3 + b_2 a_4 b_3)^*$ над $\widehat{\Sigma} = \{a_1, a_2, a_3, a_4, b_1, b_2, b_3\}$. Далее введем

$$First(\hat{\alpha}) = \{x \in \widehat{\Sigma} | x\widehat{\Sigma}^* \cap L(\hat{\alpha}) \neq \varnothing\} \tag{3.1}$$

$$Last(\hat{\alpha}) = \{x \in \widehat{\Sigma} | \widehat{\Sigma}^* x \cap L(\hat{\alpha}) \neq \varnothing\} \tag{3.2}$$

$$Follow(\hat{\alpha}) = \{(x, y) \in \widehat{\Sigma}^2 | \widehat{\Sigma}^* xy \widehat{\Sigma}^* \cap L(\hat{\alpha}) \neq \varnothing\} \tag{3.3}$$

Говоря проще, $First(\hat{\alpha})$ — множество первых букв слов языка $L(\hat{\alpha})$, $Last(\hat{\alpha})$ — множество последних букв слов языка $L(\hat{\alpha})$, а $Follow(\hat{\alpha})$ — множество пар букв, идущих друг за другом.

Построим следующий ДКА $\widehat{Aut}_\alpha = (\widehat{Q}, \widehat{\Sigma}, \widehat{Start}, \widehat{Final}, \widehat{T})$, в котором

- $\widehat{Q} = \{[\epsilon]\} \cup \{[x] | x \in \widehat{\Sigma}\} \cup \{\mathbf{T}\}$ — нестартовые нетупиковые состояния помечены буквами нового алфавита;
- $\widehat{\Sigma}$ — модифицированный алфавит;
- $\widehat{Start} = [\epsilon]$ — начальное состояние;
- $\widehat{Final} = \{[x] | x \in Last(\hat{\alpha})\}$ — финальные состояния помечены буквами, на которые могут заканчиваться слова из $L(\hat{\alpha})$;
- $\widehat{T}$ — таблица переходов вида

$$\widehat{T}([\epsilon], x) = \begin{cases} [x], & x \in First(\hat{\alpha}) \\ \mathbf{T}, & x \notin First(\hat{\alpha}) \end{cases}, \qquad \widehat{T}([x], y) = \begin{cases} [y], & x \in Follow(\hat{\alpha}) \\ \mathbf{T}, & x \notin Follow(\hat{\alpha}) \end{cases}, \qquad \forall x \in \Sigma\, \widehat{T}(\mathbf{T}, x) = \mathbf{T}$$

Тогда назовем *автоматом Глушкова* следующий НКА $Aut_\alpha = (\widehat{Q}, \Sigma, \widehat{Start}, \widehat{Final}, T)$, в котором

$$\forall i \in [1; N],\ r \in [1; \#_{a_i}(\alpha)] \qquad T(q_1, a_i) = q_2 \iff \widehat{T}(q_1, a_{ir}) = q_2$$

Фактически Aut_α получен $\widehat{Aut}_\alpha$ «стиранием» всех порядковых номеров букв, то есть заменой a_{ir} на a_i.

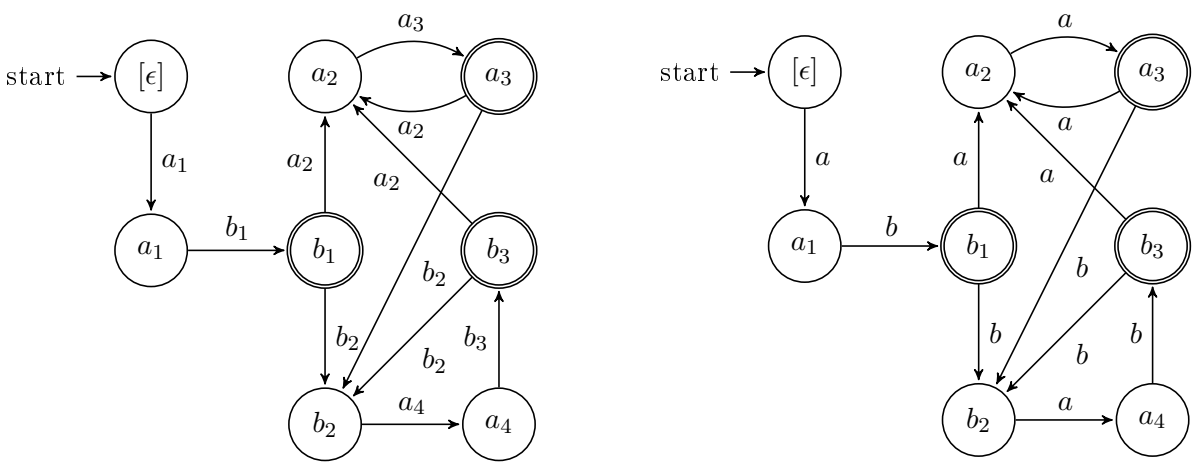
Пример 20. Построим автомат Глушкова для упомянутого выше выражения $\alpha = ab(a^2 + bab)^*$. Как уже было сказано, линеаризованное выражение имеет вид $\hat{\alpha} = a_1 b_1 (a_2 a_3 + b_2 a_4 b_3)^*$ над $\widehat{\Sigma} = \{a_1, a_2, a_3, a_4, b_1, b_2, b_3\}$. Далее имеем

$$First(\hat{\alpha}) = \{a_1\} \quad (3.4)$$

$$Last(\hat{\alpha}) = \{b_1, a_3, b_3\} \quad (3.5)$$

$$Follow(\hat{\alpha}) = \{(a_1, b_1), (b_1, a_2), (b_1, b_2), (a_2, a_3), (b_2, a_4), (a_4, a_3), (a_3, b_2), (a_3, a_2), (b_3, a_2), (b_3, b_2)\} \quad (3.6)$$

Слева построен ДКА по линеаризованному регулярному выражению. Справа — соответствующий автомат Глушкова. [Переходы в тупиковое состояние не изображены для простоты картинки.]



... [АЛГОРИТМ ГЛУШКОВА] ...

Следующая теорема объясняет, почему конструкция Глушкова работает.

Теорема 3.2. *Язык автомата Глушкова* Aut_α *есть* $L(\alpha)$.

Доказательство. Покажем сперва, что $L(\hat{\alpha}) = L(\widehat{Aut_\alpha})$. Пусть $S \subset \widehat{\Sigma}$, и $\widehat{S}$ — соответствующее подмножество состояний автомата $\widehat{Aut_\alpha}$; тогда для любых $x, y \in \widehat{\Sigma}$ верно

$$L(\alpha^{\widehat{S}}_{[x],[y]}) = \{w \in \widehat{\Sigma}^* | w \in \widehat{\Sigma}^* y \bigcap_{s \in S} \widehat{\Sigma}^* s \widehat{\Sigma}^*,\ \forall s \in S \cup \{x,y\}\ w \in s\widehat{\Sigma}^* \Rightarrow (x,s) \in Follow(\hat{\alpha}),\ \forall s_1, s_2 \in S \cup \{x,y\}\ w \in \widehat{\Sigma}$$

Иными словами, $\alpha^{\widehat{S}}_{[x],[y]}$ описывает те и только те слова, которые содержат буквы из S, начинаются в x, заканчиваются в y, и любые две соседние буквы могут быть соседними буквами в словах из $L(\hat{\alpha})$. Это можно доказать индукцией по $|S|$. База очевидна: язык $L(\alpha^{\varnothing}_{[x],[y]})$ будет пустым, если x и y не могут соседствовать, и будет $\{y\}$, если существуте переход $[x] \xrightarrow{y} [y]$ (других переходов между такими состояниями нет). Шаг доказывается аналогично теоерме ...: для некоторой $z \in S$ имеем

$$\alpha^{\widehat{S}}_{[x],[y]} = \alpha^{\widehat{S}-\{z\}}_{[x],[y]} + \alpha^{\widehat{S}-\{z\}}_{[x],[z]} (\alpha^{\widehat{S}-\{z\}}_{[z],[z]})^* \alpha^{\widehat{S}-\{z\}}_{[z],[y]}$$

С другой стороны,

$$\{w \in \widehat{\Sigma}^* | w \in \widehat{\Sigma}^* y \bigcap_{s \in S} \widehat{\Sigma}^* s \widehat{\Sigma}^*,\ \forall s_1, s_2 \in S \cup \{x,y\}\ w \in \widehat{\Sigma}^* s_1 s_2 \widehat{\Sigma}^* \Rightarrow (s_1, s_2) \in Follow(\hat{\alpha})\} = C_z \cup N_z,$$

где C_z — искомые слова, содержащие z, а N_z — искомые слова, не содержащие z:

$$C_z = \{w \in \widehat{\Sigma}^* | w \in \widehat{\Sigma}^* y \bigcap_{s \in S} \widehat{\Sigma}^* s \widehat{\Sigma}^*,\ w \in \widehat{\Sigma}^* z \widehat{\Sigma}^*,\ \forall s_1, s_2 \in S \cup \{x,y\}\ w \in \widehat{\Sigma}^* s_1 s_2 \widehat{\Sigma}^* \Rightarrow (s_1, s_2) \in Follow(\hat{\alpha})\} \quad (3.7)$$

$$N_z = \{w \in \widehat{\Sigma}^* | w \in \widehat{\Sigma}^* y \bigcap_{s \in S} \widehat{\Sigma}^* s \widehat{\Sigma}^*,\ w \notin \widehat{\Sigma}^* z \widehat{\Sigma}^*,\ \forall s_1, s_2 \in S \cup \{x,y\}\ w \in \widehat{\Sigma}^* s_1 s_2 \widehat{\Sigma}^* \Rightarrow (s_1, s_2) \in Follow(\hat{\alpha})\} \quad (3.8)$$

Тогда $N_z = L(\alpha^{\widehat{S}-\{z\}}_{[x],[y]})$ по предположению, а $C_z = L(\alpha^{\widehat{S}-\{z\}}_{[x],[z]} (\alpha^{\widehat{S}-\{z\}}_{[z],[z]})^* \alpha^{\widehat{S}-\{z\}}_{[z],[y]})$. Действительно,

$$C_z = \{w_1 w_2 w_3 | w_1, w_3 \notin \widehat{\Sigma}^* z \widehat{\Sigma}^*,\ w_2 \in z\widehat{\Sigma}^* z \cup \{\epsilon\}, w_3 \in \widehat{\Sigma}^* y\, \forall s_1, s_2 \in S \cup \{x,y\}\ w_1 w_2 w_3 \in \widehat{\Sigma}^* s_1 s_2 \widehat{\Sigma}^* \Rightarrow (s_1, s_2) \in Fo$$

Язык слов w_1 есть $L(\alpha^{\widehat{S}-\{z\}}_{[x],[z]})$, слова w_2 образуют $L(\alpha^{\widehat{S}-\{z\}}_{[z],[z]})^*$, а слова w_3 образуют $\alpha^{\widehat{S}-\{z\}}_{[z],[y]}$.

Несложно заметить, что при добавлении состояния $\mathbf{T}$ язык не меняется. Для выражений $\alpha^{\widehat{S}}_{[\epsilon],[y]}$ же имеем вместо условия $\forall s \in S \cup \{x,y\}\ w \in s\widehat{\Sigma}^* \Rightarrow (x,s) \in Follow(\hat{\alpha})$ условие $\forall s \in S \cup \{x,y\}\ w \in s\widehat{\Sigma}^* \Rightarrow s \in First(\hat{\alpha})$; доказательство аналогично.

Теперь докажем, что $L(Aut_\alpha) = L(\alpha)$. Во-первых, если $\widehat{w} = \prod_{j=1}^{|w|} a_{i_l j_l} \in L(\widehat{Aut_\alpha})$, то $w = \prod_{j=1}^{|w|} a_{i_l} \in L(Aut_\alpha)$: индукцией по $|w|$ и $|\widehat{w}|$ можно убедиться, что

$$\forall q_1, q_2 \in \widehat{Q} \qquad q_1 \xrightarrow{\widehat{w}}_{\widehat{Aut_\alpha}} q_2 \Rightarrow q_1 \xrightarrow{w}_{Aut_\alpha} q_2.$$

Во-вторых, если $w = \prod_{j=1}^{|w|} a_{i_l} \in L(Aut_\alpha)$ и $w \neq \epsilon$, то пусть $[s_1], \ldots, [s_{|w|}]$ — последовательность состояний, пройденных при прочтении w, то есть

$$[\epsilon] \longrightarrow [s_1] \longrightarrow \cdots \longrightarrow [s_{|w|}]$$

Тогда $s_1 \ldots s_{|w|} \in L(\widehat{Aut_\alpha})$, следовательно, $s_1 \ldots s_{|w|} \in L(\hat{\alpha})$. Отсюда сразу же следует, что $w \in L(\alpha)$. Тем самым, мы доказали, что $L(Aut_\alpha) = L(\alpha)$. □

Задача 3.21. Используя теорему ..., постройте другой алгоритм построения НКА по регулярному выражению и докажите его корректность. Верно ли, что полученный автомат будет совпадать с автоматом Глушкова? Этот алгоритм мы будем называть *алгоритмом Томпсона*.

Воспользовавшись идеей Глушкова, мы можем построить алгоритм построения ДКА по регулярному выражению. Следуя [?asethiu], пронумеруем по порядку все буквы регулярного выражения, введем аналоги $First(\cdot), Last(\cdot), Follow(\cdot)$ и с помощью них построим автомат, состояниями которого будут подмножества позиций в регулярном выражении.

Сначала формально определим ДКА $\overline{Aut}_\alpha$, который будем строить, затем предъявим строящий его алгоритм. Итак, пусть α — регулярное выражение над Σ, определим $\overline{\alpha}\# \in (\Sigma \cup \#)^*$ как слово, полученное из α стиранием всех скобок и знаков операций и добавлением $\#$ в конец, и $S = [1, \ldots, |\overline{\alpha}|] \subset \mathbb{N}$. Определим функции $firstpos, lastpos : REG \to 2^S$ и $followpos : S \to S$:

$$firstpos(\alpha) = \{i \in S | \overline{\alpha}[i]\Sigma^* \cap L(\alpha) \neq \varnothing\} \tag{3.9}$$

$$lastpos(\alpha) = \{i \in S | \Sigma^*\overline{\alpha}[i] \cap L(\alpha) \neq \varnothing\} \tag{3.10}$$

$$\forall i \neq |\overline{\alpha}| \ followpos(i) = \{j \in S | \Sigma^*\overline{\alpha}[i]\overline{\alpha}[j]\Sigma^* \cap L(\alpha) \neq \varnothing\}, follow(|\overline{\alpha}|) = \varnothing \tag{3.11}$$

Здесь же становится ясно, зачем мы добавили $\#$: в отличие от автомата Глушкова здесь нет выделенного стартового состояния [не заданного позицией в регулярном выражении], поэтому нужно выделить отдельно финальные состояния, в которые совершается переход по буквам $lastpos(\alpha)$. За решеткой не следует никакого символа, поэтому $follow(|\overline{\alpha}|) = \varnothing$.

Определим $\overline{Aut}_\alpha = (\overline{Q}, \Sigma, \overline{Start}, \overline{Final}, \overline{T})$, в котором

- $\overline{Q} = 2^S \cup \{\mathbf{T}\}$ — подмножества множества буквенных позиций регулярного выражения α;
- Σ — оригинальный алфавит;
- $\overline{Start} = firstpos(\alpha)$ — множество позиций первых букв $L(\alpha)$;
- $\overline{Final} = \{A \in 2^S \mid lastpos(\alpha) \subset A\}$ — финальные состояния помечены позициями, среди которых есть позиции последних букв слов из $L(\alpha)$;
- $\overline{T}$ — таблица переходов вида

$$\forall A \in 2^S \ \overline{T}(A, x) = \begin{cases} \{followpos(i) \mid i \in A, x = \overline{\alpha}[i]\}, & \text{если непусто} \\ \mathbf{T}, & \text{иначе} \end{cases}, \quad \forall x \in \Sigma \ \overline{T}(\mathbf{T}, x) = \mathbf{T}$$

Задача 3.22. Проверьте, что $L(\overline{Aut}_\alpha) = L(\alpha)$.

Теперь предъявим алгоритм для его построения. Функции $firstpos(\cdot)$, $lastpos(\cdot)$ и $followpos(\cdot)$ можно вычислить рекурсивным обходом по синтаксическому дереву регулярного выражения. В листьях стоят атомарные регулярные выражения, в остальных узлах — операции $+$, $\cdot$ и $*$. Например, для выражения $a(a+b)^*ba^*b$ имеем выражение на ПИКЧА.

Каждый узел v синтаксического дерева можно рассматривать как корень поддерева, являющегося синтаксическим деревом некоторого регулярного выражения β_v. Поэтому функции $firstpos(\cdot)$ и $lastpos(\cdot)$ будем вычислять на вершинах дерева: $firstpos(v)$ зададим как $firstpos(\beta_v)$, аналогично определим $lastpos(v)$. Для их вычисления дополнительно введем функцию $nullable : REG \to \{0, 1\}$, равную 1, если в языке регулярного выражения есть ϵ, и 0 в противном случае. Все три функции вычисляем реккурентно по следующим правилам:

...

Тогда $firstpos(\alpha)$ и $lastpos(\alpha)$ получаются как значения в корне дерева.

Зная значения $firstpos(\cdot)$ и $lastpos(\cdot)$, вычислим $followpos(\cdot)$. Ее можно вычислить, обходя в ширину синтаксическое дерево и в каждой вершине v проделывая следующие операции

- если в v стоит операция $\cdot$, а ее потомки есть v_1 слева и v_2 справа, то

$$\forall i \in lastpos(v_1) \quad followpos(i) += firstpos(v_2)$$

- если стоит операция $*$, а ее потомок есть u, то

$$\forall i \in lastpos(u) \quad followpos(i) += firstpos(u)$$

- в остальных случаях ничего не происходит.

Теперь можно смело строить ДКА. Достаточно рассматривать лишь состояния, достижимые из $firstpos(\alpha)$.

1. Построить синтаксическое дерево выражения α.
2. Начиная с листьев, реккурентно по дереву вычислить $firstpos(\cdot)$ и $lastpos(\cdot)$, используя таблицу ...
3. Обходя в ширину синтаксичесекое дерево, вычислить $followpos(\cdot)$.
4. Отметить $firstpos(\alpha) \in 2^S \cup \mathbf{T}$ как стартовое состояние, пометить его.
5. Пока существует помеченное состояние с непомеченным соседом [то есть имеется переход $q \xrightarrow{x} \overline{T(q,x)}$, в котором q помечено, а $\overline{T(q,x)}$ – нет] — пометить этого соседа.
6. Финальные состояния — все подмножества S, содержащие $lastpos(\alpha)$.

Пример 21. Построим согласно приведенному алгоритму ДКА по РВ $a(a+b)^*ba^*b$. На картинке ПИКЧА показано синтаксическое дерево и результаты вычисления $firstpos(\cdot)$ и $lastpos(\cdot)$. Вершины, для которых *nullable* равен 1, мы выделили квадратиком.

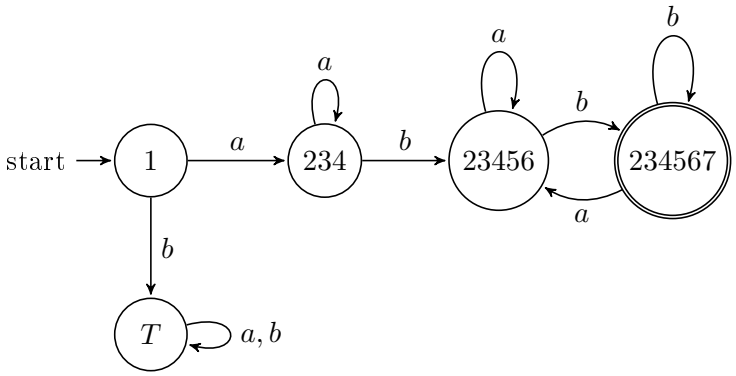
Теперь вычислим $followpos(\cdot)$. В корне дерева имеем $followpos(6) = 7$. В его левом потомке получаем $followpos(4) = followpos(5) = 6$. На следующем уровне имеем $followpos(4) = followpos(5) = 5$. Далее имеем $followpos(1) = followpos(2) = followpos(3) = 4$. На последних уровнях добавляем к $followpos(1), followpos(2), followpos(3)$ подмножество $\{2,3\}$. Итоговые значения $followpos(\cdot)$ записаны в табличке:

...

Стартовое состояние ДКА равно $firstpos(a(a+b)^*ba^*b) = \{1\}$, по a совершается переход в $followpos(1) = \{2,3,4\}$, а по b — в состояние $\mathbf{T}$: никакие слова данного языка не начинаются на b. В позициях 2, 3, 4 a стоит на месте 2, а b — на местах 3 и 4, поэтому

$$\{2,3,4\} \xrightarrow{a} followpos(2) = \{2,3,4\}; \quad \{2,3,4\} \xrightarrow{b} followpos(3) \cup followpos(4) = \{2,3,4,5,6\}$$

Аналогично достраиваем остальные переходы, имеем ДКА на следующей картинке.



Задача 3.23. **(a)** Проверьте, что приведенный выше алгоритм корректно строит $\overline{Aut}_\alpha$.

(b) Несложно убедиться в том, что в построенном по α ДКА всего $O(2^{|\alpha|})$ состояний. Предъявите семейство регулярных выражений α_n таких, что $|\alpha_n| = \Theta(n)$, а число состояний в $\overline{Aut}_{\alpha_n}$ равно $\Omega(2^n)$.

Построение конечного автомата по регулярному выражению помогает решить задачу *сопоставления образцов*[1]: дано слово w и регулярное выражение α, верно ли, что $w \in L(\alpha)$? Достаточно построить конечный автомат Aut_α любыми из вышеперечисленных алгоритмов и проэмулировать его работу на w. Аргументом за использование того или иного алгоритма является время его работы и размер используемой памяти.

[1] pattern matching

Задача 3.24. Оцените асимптотически сложность [в худшем случае]

(a) каждого из трех алгоритмов построения конечного автомата по регулярному выражению как $O\big(f(|\alpha|)\big)$;

(b) время работы каждого из построенных автоматов на входном слове как $O\big(f(|w|, |\alpha|)\big)$.

Подробнее об этом, а также о других использованиях регулярных языков в работе с текстами мы поговорим в главе ...

Алгебры Клини и уравнения с регулярными коэффициентами

Сформулируем алгебраически свойства, которыми мы пользуемся при работе с регулярными выражениями. Это даст еще один способ интерпретации регулярных языков.

Определение. *Алгебра Клини* $\mathcal{K}$ — множество с операциями $+$, $\cdot$, $*$, удовлетворяющими аксиомам: (здесь a, b, c - любые элементы множества)

- $a + (b + c) = (a + b) + c$, $a(bc) = (ab)c$
- $a + b = b + a$
- $a + a = a$
- $a + 0 = a$
- $a \cdot 1 = 1 \cdot a = a$, $a \cdot 0 = 0 \cdot a = 0$
- $a(b + c) = ab + ac$, $(a + b)c = ac + bc$
- $1 + aa^* = 1 + a^*a = a^*$

Кроме того введем на $\mathcal{K}$ отношение $\leqslant$ таким образом

$$a \leqslant b \iff a + b = b$$

и потребуем выполнения следующих аксиом:

- $b + ac \leqslant c \iff a^*b \leqslant c$
- $b + ca \leqslant c \iff ba^* \leqslant c$

Пусть Σ^* — алгебра Клини регулярных выражений, тогда квадратные матрицы с коэффициентами из регулярных выражений образуют алгебру Клини $Mat(n, \Sigma^*)$. Операции $+, \cdot$ — линейно алгебраические, а $*$ вводится как

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^* = \sum_{n=0}^{\infty} \begin{bmatrix} a & b \\ c & d \end{bmatrix}^n, \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix}^0 = \begin{bmatrix} \epsilon & 0 \\ 0 & \epsilon \end{bmatrix}$$

Покажем, что для любой такой матрицы $A^* = I + A + A^2 + \ldots$, где I — диагональная матрица с ϵ на диагонали, а умножение понимается в линейно-алгебраическом смысле. Действительно, в матрице A^n стоят слова от a, b, c, d длины n, покажем, что

$$A^n = \begin{bmatrix} [(a + bd^*c)^*]_n & [(a + bd^*c)^*bd^*]_n \\ [(d + ca^*b)^*ca^*]_n & [(d + ca^*b)^*]_n \end{bmatrix}$$

где $[L]_n$ — слова, содержащие суммарно n букв a, b, c, d. Это доказывается по индукции, база при $n = 1$ очевидна. Теперь рассмотрим $A^{n+1} = A^n \cdot A$:

$$\begin{bmatrix} [(a + bd^*c)^*]_n & [(a + bd^*c)^*bd^*]_n \\ [(d + ca^*b)^*ca^*]_n & [(d + ca^*b)^*]_n \end{bmatrix} \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix} =$$

$$= \begin{bmatrix} [(a+bd^*c)^*]_n a + [(a+bd^*c)^* bd^*]_n c & [(a+bd^*c)^*]_n b + [(a+bd^*c)^* bd^*]_n d \\ [(d+ca^*b)^* ca^*]_n a + [(d+ca^*b)^*]_n c & [(d+ca^*b)^* ca^*]_n b + [(d+ca^*b)^*]_n d \end{bmatrix}$$

Что имеем покомпонентно? Любое слово из $[(a+bd^*c)^*]_{n+1}$ оканчивается либо на a (все такие слова есть очевидно $[(a+bd^*c)^*]_n a$), либо на c (все такие образуют $[(a+bd^*c)^* bd^*]_n c$), тогда $[(a+bd^*c)^*]_n a + [(a+bd^*c)^* bd^*]_n c = [(a+bd^*c)^*]_{n+1}$. Аналогично рассматриваются остальные элементы матрицы A^{n+1}: надо просто посмотреть, на какие буквы заканчиваются слова из $[(a+bd^*c)^* bd^*]_{n+1}$, $[(d+ca^*b)^* ca^*]_{n+1}$, $[(d+ca^*b)^*]_{n+1}$.

Для матриц $N \times N$ сделаем следующее: разобьем матрицу на блоки и определим

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^* = \begin{bmatrix} (A+BD^*C)^* & (A+BD^*C)^* BD^* \\ (D+CA^*B)^* CA^* & (D+CA^*B)^* \end{bmatrix}$$

Эта конструкция определена корректно: здесь звезда Клини применется лишь к прямоугольным матрицам.

Это поможет в поиске решений линейных уравнений от регулярных функций. Пусть есть регулярные выражения a_{ij}, b_j, переменные x_i и система уравнений $x_i = \sum a_{ij} x_j + b_j$. Система может быть записана как $x = Ax + b$, где $x = (x_i), A = (a_{ij}), b = (b_i)$.

Теорема 3.3. *Пусть дана линейная система $x_i = \sum a_{ij} x_j + b_j$, где $a_{ij}, b_j \in \mathcal{K}$, $x = (x_i)$ — вектор из x_i в линейно-алгебраическом смысле этого слова, а $A = (a_{ij})$ — матрица из регулярных выражений. Тогда $x = A^* b$ — минимальное по включению решение.*

Задача 3.25. Докажите эту теорему.

Вывод. Если a, b — регулярные выражения, то $a^* b$ — наименьшее по включению решение уравнения $x = ax + b$.

Задача 3.26. В условиях данного следствия докажите, что если $\epsilon \notin a$, то минимальное решение также является единственным.

Задача 3.27. Найдите все решения уравнения $x = ax + b$, если $\epsilon \in a$.

Решение данной системы можно интерпретировать в терминах конечных автоматов. $x = Ax + b$, a_{ij}, b_i — регулярные выражения, для них есть НКА A_{ij}, B_i. Заведем состояния $s_1, \ldots, s_n, f$, где s_i соответствует переменной x_i, а f — финальное — соответствует свободному члену.

«Вклеим» автоматы A_{ij} между s_i и s_j и B_i между s_i и f. Тогда x_i есть просто слова, которые можно прочитать по пути из s_i в f, проходя через $s_1, s_2, \ldots, s_n$.

ПИКЧА

Обобщения регулярных выражений

Мы можем также рассматривать регулярные выражения, использующие дополнительные операции, вроде $\cap$ [пересечение] и $\complement$ [дополнение]. Введем расширенные регулярные выражения, то есть РВ, использующие дополнения.

Определение. Множество *расширенных регулярных выражений xREG* над алфавитом Σ — множество строчек над алфавитом $\Sigma \cup \{+, \cdot, *, (,), \complement\}$, удовлетворяющее следующим правилам:

- $\varnothing, \epsilon, x$ для любой $x \in \Sigma$ — *атомарные* регулярные выражения;
- если α и β — регулярные выражения, то $\alpha + \beta, \alpha\beta, \alpha^*, (\alpha), \alpha^\complement$ — расширенные регулярные выражения;
- никаких других расширенных регулярных выражений нет, то есть любое расширенное регулярное выражение может быть получено из атомарных с помощью операций $+, \cdot, *, \complement$ и использования скобок.

Язык $L(\alpha)$ расширенного регулярного выражения α определяется так же, как и для обычного регулярного выражения, с тем дополнительным условием, что $L(\alpha^{\complement}) = L(\alpha)^{\complement}$. По формулам де Моргана $\alpha \cap \beta = (\alpha^{\complement} + \beta^{\complement})^{\complement}$, поэтому пересечение как операция выражается через остальные.

Определение. Регулярный язык $L \subset \Sigma^*$ назовем *$*$-свободным*[2], если он может быть задан расширенным регулярным выражением без звезды Клини.

Заметим, что $\varnothing^{\complement} = \Sigma^*$. Таким образом, в частности, язык всех слов, содержащих подслово $w \in \Sigma^*$, является $*$-свободным: он может быть задан регулярным выражением $\varnothing^{\complement} w \varnothing^{\complement}$.

Задача 3.28. Покажите, что $(ab)^*$ $*$-свободен.

Решение. Построим для $(ab)^*$ расширенное регулярное выражение, не использующее звездочку Клини. Непустое слово лежит в данном языке титтк выполняются следующие условия:

- оно начинается на a и заканчивается на b;
- за каждой буквой a следует буква b;
- за каждой буквой b следует буква a.

Слова, удовлетворяющие первому условию, образуют язык $a\varnothing^{\complement}b$. Слова, удовлетворяющие второму условию, не содержат подслова a^2, следовательно, описываются расширенным РВ $(\varnothing^{\complement} a^2 \varnothing^{\complement})^{\complement}$. Язык слов, удовлетворяющих третьему условию, есть по аналогии $(\varnothing^{\complement} b^2 \varnothing^{\complement})^{\complement}$. Искомый язык есть пересечение всех этих трех языков; так как они $*$-свободны, то и $(ab)^*$ $*$-свободен. □

Между тем для языка $(a^2)^*$ не существует $*$-свободного расширенного регулярного выражения. Шютценберже сформулировал критерии $*$-свободности языка; к нему мы вернемся позднее.

Задача 3.29. $\Sigma = \{a_0, \dots a_n\}$. Рассмотрим язык L_n, состоящий из слова $(\dots (a_0^2 a_1)^2 \dots a_n)^2$.

- Покажите, что любое регулярное выражение, задающее L_n, имеет длину $\Omega(2^n)$.
- Постройте регулярное выражение *с операцией пересечения*, имеющее длину $O(n^2)$.

Решение.

- Решение почти аналогично решению задачи XXX: кратчайшее РВ, задающее данный язык, не может содержать звезды Клини и, следовательно, содержит по разу все буквы слова $(\dots (a_0^2 a_1)^2 \dots a_n)^2$, коих очевидно $\Omega(2^n)$.
- В прошлом пункте нельзя было использовать звездочку Клини, здесь же можно: данный язык можно представить как пересечение некоторых бесконечных языков. Если точнее, мы получим $(\dots (a_0^2 a_1)^2 \dots a_n)^2$ как пересечение $n+1$ языка, каждый из которых записывается регулярным выражением длины не более Cn для константы $C \in \mathbb{N}$.

  Для всех $k \in [1; n]$ рассмотрим языки

  $$L_k := (\dots ((((a_0 + \dots a_{k-1})^* a_k)^2 a_{k+1})^* a_{k+2})^* \dots a_n)^*$$

  и введем также $L_0 = (\dots ((a_0^2 a_1)^* a_2)^* \dots a_n)^*$. Докажем, что $(\dots (a_0^2 a_1)^2 \dots a_n)^2 = \bigcap_{i \in [0;n]} L_i$. Слово лежит в $L_n = ((a_0 + \dots a_{n-1})^* a_n)^2$ титтк оно имеет вид $w_1 a_n w_2 a_n$, где $w_1, w_2 \in \{a_0, \dots a_{n-1}\}$. Это самое слово должно лежать в $L_{n-1} = (((a_0 + \dots a_{n-2})^* a_{n-1})^2 a_n)^*$, тогда

  $$w_1 a_n w_2 a_n \in L_{n-1} \iff w_1 = u_1 a_{n-1} u_2 a_{n-1}, w_2 = u_3 a_{n-1} u_4 a_{n-1}$$

  Продолжая по аналогии для $L_{n-2}, \dots L_0$, получаем, что в пересечении всех языков L_i лежит ровно одно слово $(\dots (a_0^2 a_1)^2 \dots a_n)^2$. При этом длина регулярного выражения L_k равна

  $$\underbrace{2(2k+3)}_{\text{длина } (a_0+\dots a_{k-1})^* a_k)^2} + \underbrace{2(n-k)}_{\text{остальные буквы и операции}} + \underbrace{2(n-k-1)}_{\text{скобки}} = 4n+4$$

  , таким образом, длина всего выражения есть $(4n+4)(n+1)+n = O(n^2)$, как и требовалось. □

[2] star-free по-английски