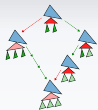


Регулярные грамматики и выражения. Теорема Клини



Теория формальных языков
2023 г.



Грамматики

Определение

Грамматика — это четвёрка $G = \langle N, \Sigma, P, S \rangle$, где:

- N — алфавит нетерминалов;
- Σ — алфавит терминалов;
- P — множество правил переписывания $\alpha \rightarrow \beta$ типа $\langle (N \cup \Sigma)^+ \times (N \cup \Sigma)^* \rangle$;
- $S \in N$ — начальный символ.

$\alpha \rightarrow \beta$, если $\alpha = \gamma_1 \alpha' \gamma_2$, $\beta = \gamma_1 \beta' \gamma_2$, и $\alpha' \rightarrow \beta' \in P$.
 \rightarrow^* — рефлексивное транзитивное замыкание \rightarrow .

Язык $\mathcal{L}(G)$, порождаемый G — множество $\{u \mid u \in \Sigma^* \text{ \& } S \Rightarrow^* u\}$. Сентенциальная форма — элемент множества $\{u \mid u \in (N \cup \Sigma)^* \text{ \& } S \Rightarrow^* u\}$.



Регулярные грамматики и НКА

Регулярная (праволинейная) грамматика G содержит правила вида $S \rightarrow \varepsilon$ (причём S не встречается в правых частях никаких правил), $T_i \rightarrow a_i$, $T_i \rightarrow a_i T_j$.

То есть во всех сентенциальных формах либо нет нетерминалов, либо он единствен и расположен строго справа от терминальных символов.

Каждый нетерминал N описывает собственный язык $\mathcal{L}(N)$ относительно G — язык слов, которые выводятся из N за конечное число применений правил грамматики G .



Регулярные грамматики и НКА

Регулярная (праволинейная) грамматика G содержит правила вида $S \rightarrow \varepsilon$ (причём S не встречается в правых частях никаких правил), $T_i \rightarrow a_i$, $T_i \rightarrow a_i T_j$.

То есть во всех сентенциальных формах либо нет нетерминалов, либо он единствен и расположен строго справа от терминальных символов.

НКА (неформально) определяется списком правил перехода и финальными состояниями.

- $T_i \rightarrow a_i T_j$ соответствует переходу $\langle T_i, a_i, T_j \rangle$;
- $T_i \rightarrow a_i$ соответствует переходу $\langle T_i, a_i, F \rangle$, где F — уникальное финальное состояние;
- $S \rightarrow \varepsilon$ соответствует объявлению S финальным.



Операции в регулярных грамматиках

Объединение

Дано: G_1 и G_2 — праволинейные. Построить $G : \mathcal{L}(G) = \mathcal{L}(G_1) \cup \mathcal{L}(G_2)$.

- 1 Переименовать нетерминалы из N_1 и N_2 , чтобы стало $N_1 \cap N_2 = \emptyset$ (сделать α -преобразование). Применить переименовку к правилам G_1 и G_2 .
- 2 Объявить стартовым символом свежий нетерминал S и для всех правил G_1 вида $S_1 \rightarrow \alpha$ и правил G_2 вида $S_2 \rightarrow \beta$, добавить правила $S \rightarrow \alpha$, $S \rightarrow \beta$ в правила G .
- 3 Добавить в правила G остальные правила из G_1 и G_2 .



Операции в регулярных грамматиках

Конкатенация

Дано: G_1 и G_2 — праволинейные. Построить $G : \mathcal{L}(G) = \mathcal{L}(G_1) \mathcal{L}(G_2)$.

- 1 Переименовать нетерминалы из N_1 и N_2 , чтобы стало $N_1 \cap N_2 = \emptyset$ (сделать α -преобразование).
- 2 Построить из G_1 её вариант без ε -правил (см. ниже).
- 3 По всякому правилу из G_1 вида $A \rightarrow a$ строим правило G вида $A \rightarrow aS_2$, где S_2 — стартовый нетерминал G_2 .
- 4 Добавить в правила G остальные правила из G_1 и G_2 .
Объявить S_1 стартовым.
- 5 Если $\varepsilon \in \mathcal{L}(G_1)$ (до шага 2), то по всем $S_2 \rightarrow \beta$ добавить правило $S_1 \rightarrow \beta$.



Операции в регулярных грамматиках

Положительная итерация Клини

Дано: G_1 — праволинейная. Построить

$G : \mathcal{L}(G) = \mathcal{L}(G_1)^+$.

- 1 Построить из G_1 её вариант без ε -правил.
- 2 По всякому правилу из G_1 вида $A \rightarrow a$ строим правило G вида $A \rightarrow aS_1$, где S_1 — стартовый нетерминал G_1 .
- 3 Добавить в правила G все (включая вида $A \rightarrow a$) правила из G_1 . Объявить S_1 стартовым.
- 4 Если $\varepsilon \in \mathcal{L}(G_1)$ (до шага 2), добавить правило $S_1 \rightarrow \varepsilon$ и вывести S_1 из рекурсии.



Построение грамматики без ε -правил

Дано: G — праволинейная. Построить G' без правил вида $A \rightarrow \varepsilon$ такую, что $\mathcal{L}(G') = \mathcal{L}(G)$ или $\mathcal{L}(G') \cup \{\varepsilon\} = \mathcal{L}(G)$.

- 1 Перенести в G' все правила G , не имеющие вид $A \rightarrow \varepsilon$.
- 2 Если существует правило $A \rightarrow \varepsilon$, то по всем правилам вида $B \rightarrow \alpha A$ дополнительно строим правила $B \rightarrow \alpha$.

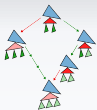


Пересечение регулярных грамматик

Дано: G_1, G_2 — праволинейные. Построить G' такую, что

$$\mathcal{L}(G') = \mathcal{L}(G_1) \cap \mathcal{L}(G_2).$$

- ❶ Построить стартовый символ G' — пару $\langle S_1, S_2 \rangle$, где S_i — стартовый символ грамматики G_i .
- ❷ Поместить $\langle S_1, S_2 \rangle$ в множество U неразобранных нетерминалов. Множество T разобранных нетерминалов объявить пустым.
- ❸ Для каждого очередного нетерминала $\langle A_1, A_2 \rangle \in U$:
 - ❶ если $A_1 \rightarrow a \in G_1, A_2 \rightarrow a \in G_2$, тогда добавить в G' правило $\langle A_1, A_2 \rangle \rightarrow a$;
 - ❷ если $A_1 \rightarrow aA_3 \in G_1, A_2 \rightarrow aA_4 \in G_2$, тогда добавить в G' правило $\langle A_1, A_2 \rangle \rightarrow a\langle A_3, A_4 \rangle$, а в U — нетерминал $\langle A_3, A_4 \rangle$, если его ещё нет в множестве T ;
 - ❸ если все пары правил, указанные выше, были обработаны, тогда переместить $\langle A_1, A_2 \rangle$ из U в T .
- ❹ Повторять шаг 3, пока множество U не пусто.
- ❺ Если $\varepsilon \in \mathcal{L}(G_1)$ & $\varepsilon \in \mathcal{L}(G_2)$, тогда добавить в G' правило $\langle S_1, S_2 \rangle \rightarrow \varepsilon$.



Лемма о накачке

Пусть n — число нетерминалов в регулярной грамматике G для языка \mathcal{L} .

Рассмотрим слово $w \in \mathcal{L}(G)$, $|w| \geq n + 1$. Оно получается применением цепочки из $n + 1$ правил \Rightarrow после применения хотя бы двух из них нетерминал в сентенциальной форме результата повторится.

$$\begin{array}{c}
 S \Rightarrow \dots \Rightarrow w_1 A \Rightarrow \dots \Rightarrow w_1 w_2 A \Rightarrow \dots \Rightarrow w_1 w_2 w_3 \\
 \underbrace{\hspace{10em}}_{\text{не больше } n+1 \text{ шага}} \\
 \Downarrow \\
 |w_1| + |w_2| \leq n + 1
 \end{array}$$

По построению, $w_3 \in \mathcal{L}(A)$ (поскольку A в конечном счёте раскрывается в w_3), и также $w_2 w_3 \in \mathcal{L}(A)$, причём $|w_2| > 0$. Кроме того, $w_1 \mathcal{L}(A) \subseteq \mathcal{L}(G)$, поскольку



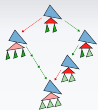
Лемма о накачке

Рассмотрим слово $w \in \mathcal{L}(G)$, $|w| \geq n + 1$. Оно получается применением цепочки из $n + 1$ правил \Rightarrow после применения хотя бы двух из них нетерминал в сентенциальной форме результата повторится.

Известно, что $|w_1| + |w_2| \leq n + 1$.

$$\begin{array}{c}
 \underbrace{S \rightarrow \dots \rightarrow w_1 A}_{\rho_1: \text{вывод } w_1 A \text{ из } S} \xrightarrow{\rho_2: \text{вывод } w_2 A \text{ из } A} \underbrace{A \rightarrow \dots \rightarrow w_1 w_2 A}_{\rho_3: \text{вывод } w_3 \text{ из } A} \rightarrow \dots \rightarrow w_1 w_2 w_3
 \end{array}$$

Поскольку $A \rightarrow^* w_2 A$, то $\forall k (A \rightarrow^* w_2^k A)$ (достаточно повторить k раз вывод ρ_2). Значит, $\forall k (w_1 w_2^k w_3 \in \mathcal{L}(G))$.



Лемма о накачке

Утверждение

Если G — регулярная, то существует такое $n \in \mathbb{N}$, что $\forall w (w \in \mathcal{L}(G) \ \& \ |w| > n \Rightarrow \exists w_1, w_2, w_3 (|w_2| > 0 \ \& \ |w_1| + |w_2| \leq n \ \& \ w = w_1 w_2 w_3 \ \& \ \forall k (k \geq 0 \Rightarrow w_1 w_2^k w_3 \in \mathcal{L}(G)))$).

Известно, что $|w_1| + |w_2| \leq n + 1$.

$$\begin{array}{c}
 \underbrace{S \rightarrow \dots \rightarrow w_1}_{\rho_1: \text{вывод } w_1 A \text{ из } S} \quad \overbrace{A \rightarrow \dots \rightarrow w_1 w_2 A}^{\rho_2: \text{вывод } w_2 A \text{ из } A} \quad \underbrace{A \rightarrow \dots \rightarrow w_1 w_2 w_3}_{\rho_3: \text{вывод } w_3 \text{ из } A}
 \end{array}$$

Поскольку $A \rightarrow^* w_2 A$, то $\forall k (A \rightarrow^* w_2^k A)$ (достаточно повторить k раз вывод ρ_2). Значит, $\forall k (w_1 w_2^k w_3 \in \mathcal{L}(G))$.



Ещё раз о структуре накачек

Если G — регулярная, то существует такое $n \in \mathbb{N}$, что $\forall w (w \in \mathcal{L}(G) \ \& \ |w| \geq n \Rightarrow \exists w_1, w_2, w_3 (|w_2| > 0 \ \& \ |w_1| + |w_2| \leq n \ \& \ w = w_1 w_2 w_3 \ \& \ \forall k (k \geq 0 \Rightarrow w_1 w_2^k w_3 \in \mathcal{L}(G))))$.

- n — длина накачки;
- w_1 — префикс накачки;
- w_2 — накачиваемый фрагмент (или просто «накачка»);
- w_3 — суффикс накачки;
- $w_1 w_2$ — область накачки;
- слово $w_1 w_3$ (случай $k = 0$) — результат «пустой накачки» или «отрицательной накачки»;
- слова $w_1 w_2^k w_3$, где $k \geq 2$ — результаты «положительной накачки».



Применение леммы о накачке

Ниже запись $x[y]$ означает, что выбор x зависит от y .

Отрицание классической леммы о накачке

Пусть \mathcal{L} — произвольный формальный язык. Если

$\forall n \in \mathbb{N} \exists w[n] (w \in \mathcal{L} \ \& \ |w| \geq n \ \& \ \forall w_1[n, w], w_2[n, w], w_3[n, w] (|w_2| > 0 \ \& \ |w_1| + |w_2| \leq n \ \& \ w = w_1 w_2 w_3 \Rightarrow \exists i[n, w, w_1, w_2, w_3] (i \geq 0 \ \& \ w_1 w_2^i w_3 \notin \mathcal{L})))$, то \mathcal{L} — не регулярный.

На розовом фоне — параметры, которые выбираются произвольно. На голубом — те, которые можно конкретизировать. Таким образом, можно трактовать применение этой формы леммы о накачке как игру двух участников: «красные» пытаются создать максимально плохие условия для её применения, а «синие» — найти выигрышную стратегию в рамках условий «красных».



Применение леммы о накачке

Ниже запись $x[y]$ означает, что выбор x зависит от y .

Отрицание классической леммы о накачке

Пусть \mathcal{L} — произвольный формальный язык. Если

$\forall n \in \mathbb{N} \exists w[n] (w \in \mathcal{L} \ \& \ |w| \geq n \ \& \ \forall w_1[n, w], w_2[n, w], w_3[n, w] (|w_2| > 0 \ \& \ |w_1| + |w_2| \leq n \ \& \ w = w_1 w_2 w_3 \Rightarrow \exists i[n, w, w_1, w_2, w_3] (i \geq 0 \ \& \ w_1 w_2^i w_3 \notin \mathcal{L})))$, то \mathcal{L} — не регулярный.

На розовом фоне — параметры, которые выбираются произвольно. На голубом — те, которые можно конкретизировать. Иногда такие игры «за кванторы» при использовании формул с большим количеством чередований \forall и \exists называют «играми демона и ангела» (d \forall emonic vs. ang \exists lic nondeterministic choice) или игрой «Абеляра и Элоизы».



Применение леммы о накачке

- Ход «красных» — выбор n . Каждое доказательство начинается фразой: «пусть n — длина накачки».
- Ход «синих»: ищем «ненакачиваемое» слово w . Это слово должно зависеть от n (его длина не меньше), и быть достаточно удобным для анализа (чтобы минимизировать количество разбиений его на фрагменты накачки).
- Ход «красных». Мы его не знаем, поэтому должны перебрать все возможные. В рамках префикса длины не больше n рассматриваем допустимые разбиения выбранного w на w_1 и w_2 . Например, если w начинается с префикса a^n , то с учётом ограничения $|w_1| + |w_2| \leq n$ возможна только ситуация, когда $w_1 = a^{k_1}$, $w_2 = a^{k_2}$, причём $k_2 \geq 1$ и $k_1 + k_2 \leq n$.
- Выбор w_1 и w_2 однозначно определяет и значение w_3 .
- Ход «синих». По каждому разбиению строим накачиваемую серию $w_1(w_2)^i w_3$ и предъявляем такое значение i_0 , что $w_1(w_2)^{i_0} w_3 \notin \mathcal{L}$.



Примеры применения леммы о накачке

Обозначим обращение (reversal) слова w как w^R . Рассмотрим язык $\mathcal{L} = \{ww^R \mid w \in \Sigma^+\}$.

Пусть длина накачки — n . Рассмотрим слово

$b^{n+1}a a b^{n+1} \in \mathcal{L}$. Поскольку $|w_1| + |w_2| \leq n$, то

$w_2 = b^k$, $k \geq 1$. Но $b^m a a b^n \notin \mathcal{L}$, если $m \neq n$. Поэтому \mathcal{L} — не регулярный.



Примеры применения леммы о накачке

Обозначим обращение (reversal) слова w как w^R . Рассмотрим язык $\mathcal{L} = \{ww^R \mid w \in \Sigma^+\}$.

Пусть длина накачки — n . Рассмотрим слово

$b^{n+1}a a b^{n+1} \in \mathcal{L}$. Поскольку $|w_1| + |w_2| \leq n$, то

$w_2 = b^k$, $k \geq 1$. Но $b^m a a b^n \notin \mathcal{L}$, если $m \neq n$. Поэтому \mathcal{L} — не регулярный.

Рассмотрим язык $\mathcal{L}' = \{a^n b^m \mid n \neq m\}$.

Пусть длина накачки — n . Рассмотрим множество слов

$a^n b^{n+n!} \in \mathcal{L}'$. Поскольку $|w_1| + |w_2| \leq n$, то

$w_2 = a^k$, $k \geq 1$. Но для всех $k \leq n \exists v(n + k \cdot v = n + n!)$, а именно $v = \frac{n!}{k}$. Поэтому слово вида $a^{n+n!} b^{n+n!} \in \mathcal{L}'$, что абсурдно. Следовательно, \mathcal{L}' не является регулярным.

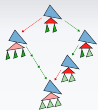


Анализ на достаточность

Является ли лемма о накачке достаточной характеристикой регулярных языков? Существуют ли языки, которые «накачиваются» согласно её формулировке, но не регулярны?

Гипотеза

G — регулярная \iff существует такое $n \in \mathbb{N}$, что $\forall w (w \in \mathcal{L}(G) \ \& \ |w| \geq n \Rightarrow \exists w_1, w_2, w_3 (|w_2| > 0 \ \& \ |w_1| + |w_2| \leq n \ \& \ w = w_1 w_2 w_3 \ \& \ \forall k (k \geq 0 \Rightarrow w_1 w_2^k w_3 \in \mathcal{L}(G)))$).



Анализ на достаточность

Гипотеза

G — регулярная \iff существует такое $n \in \mathbb{N}$, что $\forall w (w \in \mathcal{L}(G) \ \& \ |w| \geq n \Rightarrow \exists w_1, w_2, w_3 (|w_2| > 0 \ \& \ |w_1| + |w_2| \leq n \ \& \ w = w_1 w_2 w_3 \ \& \ \forall k (k \geq 0 \Rightarrow w_1 w_2^k w_3 \in \mathcal{L}(G)))$).

Рассмотрим язык $\mathcal{L} = \{w w^R z \mid w \in \Sigma^+ \ \& \ z \in \Sigma^+\}$ и $n = 4$.

- Если $|w| = 1$, тогда можно разбить слово $w w^R z$ так: $w_1 = w w^R$, $w_2 = z[1]$, $w_3 = z[2..|z|]$. Тогда для всех k $w_1 w_2^k w_3 \in \mathcal{L}$.
- Если $|w| \geq 2$, тогда разбиваем так: $w_1 = \varepsilon$, $w_2 = w[1]$, $w_3 = w[2..|w|] w^R z$. Слова $w[2..|w|] w^R z$ и $w[1]^k w[2..|w|] w^R z$ при $k \geq 2$ также принадлежат \mathcal{L} .



Анализ на достаточность

Гипотеза

G — регулярная \iff существует такое $n \in \mathbb{N}$, что $\forall w (w \in \mathcal{L}(G) \ \& \ |w| \geq n \Rightarrow \exists w_1, w_2, w_3 (|w_2| > 0 \ \& \ |w_1| + |w_2| \leq n \ \& \ w = w_1 w_2 w_3 \ \& \ \forall k (k \geq 0 \Rightarrow w_1 w_2^k w_3 \in \mathcal{L}(G)))$).

Мы нашли длину накачки для $\{w w^R z \mid w \in \Sigma^+ \ \& \ z \in \Sigma^+\}$ (она равна 4), но язык регулярным не является.

Следовательно, лемма о накачке — только необходимое, но не достаточное условие регулярности.



Смысл леммы о накачке

Структура доказательства указывает, что длина накачки n регулярного языка \mathcal{L} не больше (возможно, меньше) числа нетерминалов в минимальной грамматике для \mathcal{L} .

Покажем, что у некоторых регулярных языков длина накачки действительно меньше, чем размер минимального НКА (или минимальной регулярной грамматики).



Смысл леммы о накачке

Рассмотрим $\mathcal{L} = a \mid b \mid (a \{a \mid b\}^* a) \mid (b \{a \mid b\}^* b)$. Если выбрать длину накачки $n = 2$, то в качестве «накачки» Ψ можно взять вторую букву слова из \mathcal{L} . Пусть G имеет два нетерминала S, T и распознаёт \mathcal{L} . Если G содержит правила $S \rightarrow aT$ и $S \rightarrow bT$ (или $S \rightarrow aS, S \rightarrow bS$), то для некоторого непустого z слова вида az и bz будут либо оба принадлежать \mathcal{L} , либо нет, чего не может быть. Значит, G содержит либо пару $S \rightarrow aT, S \rightarrow bS$, либо пару $S \rightarrow bT, S \rightarrow aS$. Рассмотрим первый случай. Тогда для некоторого непустого z имеем $az \in \mathcal{L} \Leftrightarrow b^+az \in \mathcal{L}$, что абсурдно.

Таким образом, в грамматике для \mathcal{L} должно быть больше двух нетерминалов (можно обойтись тремя).



Достаточный вариант леммы о накачке

Видно, что проблемы с языком $\{w w^R z \mid w \in \Sigma^+ \text{ \& } z \in \Sigma^+\}$ возникают из-за того, что у него очень удачный префикс: любая степень буквы, большая первой, начинается с палиндрома. Однако, если бы мы потребовали, чтобы слово из \mathcal{L} начиналось с палиндрома хотя бы длины 4, подобное рассуждение уже не привело бы к успеху.



Достаточный вариант леммы о накачке

Мы можем искать не первый повтор нетерминала в пути разбора по грамматике, а любой, если осталось разобрать ещё достаточно длинный суффикс.

$$S \rightarrow \dots \rightarrow \Phi A_0 \rightarrow \Phi \Psi' A \rightarrow \dots \rightarrow \Phi \Psi' \Psi A \rightarrow \dots \rightarrow \Phi \Psi' \Psi \Theta$$

Произвольное
число шагов

Не более m шагов
до повторения нетерминала

\mathcal{L} регулярный \Leftrightarrow существует универсальная длина накачки m такая, что $w \in \mathcal{L}$ ($|w| \geq m$) для любого $i \leq |w| - m$ может быть представлено как $\Phi \Psi' \Psi^i \Theta$, где $|\Phi| = i$, $1 \geq |\Psi| \leq m$, $|\Psi'| + |\Psi| \leq m$, причём $\forall k (\Phi \Psi' \Psi^k \Theta \in \mathcal{L})$.



Академические регулярные выражения \mathcal{RE}

- $A \mid B$ — альтернатива (вхождение слова или из A , или из B);
 - AB — конкатенация (множество слов с префиксами из A и суффиксами из B);
 - A^* — итерация Клини (0 или более конкатенаций A с собой).
-
- A^+ — положительная итерация (синтаксический сахар для выражения AA^*);
 - $A?$ — опция (синтаксический сахар для выражения $(A \mid \epsilon)$).

И менее очевидные синтаксические конструкции, такие как отрицание, положительные и отрицательные «ретроспективные» и «опережающие» проверки (моделирующие в т.ч. пересечения), сохраняющие выразительную силу регулярных языков.



Академические регулярные выражения \mathcal{RE}

- $A \mid B$ — альтернатива (вхождение слова или из A , или из B);
- AB — конкатенация (множество слов с префиксами из A и суффиксами из B);
- A^* — итерация Клини (0 или более конкатенаций A с собой).

Приоритет операций: итерация $>$ конкатенация $>$ альтернатива, то есть $ab^* \mid c^*d$ — то же, что $(a(b^*)) \mid ((c^*)d)$.

Определим $r_1 = r_2 \Leftrightarrow \mathcal{L}(r_1) = \mathcal{L}(r_2)$. Для всех $r_1, r_2, r_3 \in \mathcal{RE}$:

- операции конкатенации и альтернативы ассоциативны;
- $r_1 \mid r_2 = r_2 \mid r_1$;
- $r_1(r_2 \mid r_3) = r_1r_2 \mid r_1r_3$;
- $(r_1 \mid r_2)r_3 = r_1r_3 \mid r_2r_3$.

Как описать все возможные тождества регулярных выражений?



Полукольца

Полукольцо $\mathcal{S} = \langle \mathcal{A}, \oplus, \otimes, 0 \rangle$ над носителем \mathcal{A} — это алгебраическая структура такая, что:

- \mathcal{S} — коммутативный моноид по \oplus ;
 - \mathcal{S} — полугруппа по \otimes ;
 - 0 — это id по сложению и ноль по умножению;
 - выполнены левая и правая дистрибутивности.
-
- Регулярные выражения — идемпотентное по \oplus полукольцо с единицей (ϵ) относительно $|$ и \cdot . Нуль — пустое выражение \emptyset , не распознающее никакую строку.
 - Натуральные числа с $+$, \cdot — коммутативное полукольцо с 1.
 - Если M — множество, то $\langle 2^M, \cup, \cap, \emptyset \rangle$ — идемпотентное коммутативное полукольцо с единицей, равной M .
 - $\langle \mathbb{N} \cup \{\infty\}, \min, +, \infty \rangle$ — тропическое полукольцо.



Алгебра Клини

Для полной формализации алгебры регулярных выражений требуется ввести аксиомы для $*$. Конечной аксиоматизации для неё не существует, но можно построить полную схему аксиом.

Алгебра Клини $\langle \Sigma, |, \cdot, *, \emptyset, \varepsilon \rangle$ — идемпотентное полукольцо с единицей, удовлетворяющее следующим аксиомам:

- $x^*x + 1 = x^* = 1 + xx^*$ (аксиома развёртки)
- (формализация Саломеа, **Sal**): $\forall p, q, x ((p \mid qx = x \Rightarrow x = q^*p) \ \& \ (p \mid xq = x \Rightarrow x = pq^*))$, где q не распознаёт ε — левая и правая леммы Ардена;
- (формализация Козена, **Koz**): $\forall p, q, x ((q \mid px \leq x \Rightarrow p^*q \leq x) \ \& \ (q \mid xp \leq x \Rightarrow qp^* \leq x))$, где $x \leq y \Leftrightarrow x \mid y = y, x = y \Leftrightarrow x \leq y \ \& \ y \leq x$.



Алгебра Клини

В выводах далее используются следующие условные обозначения.

Сокращение	Аксиома
(Idm)	$x \mid x = x$
(Unfold)	$\varepsilon \mid xx^* = x^*, \varepsilon \mid x^*x = x^*$
(Dstr)	$(x \mid y)z = xz \mid yz, x(y \mid z) = xy \mid xz$
(Koz)	$q \mid px \leq x \Rightarrow p^*q \leq x,$ $q \mid xp \leq x \Rightarrow qp^* \leq x$

Применение коммутативности по альтернативе и ассоциативности, а также применение аксиом единицы в выводах не указываются.



Некоторые теоремы алгебры Клини

$$(Bsm) \quad ax = xb \Rightarrow a^*x = xb^* \quad (\text{Бисимуляция})$$

$$(Sld) \quad x(yx)^* = (xy)^*x \quad (\text{Сдвиг})$$

$$(Dnst) \quad x^*(yx^*)^* = (x \mid y)^* \quad (\text{Уплощение})$$

Законы сдвига и уплощения используются в оптимизациях регулярных событий:

- закон сдвига позволяет перестраивать циклы с поствычислениями в циклы с предвычислениями;
- закон уплощения позволяет перестраивать друг в друга вложенные циклы и циклы с условиями внутри итерации.



Полнота аксиоматики

Теорема о полноте Sal и Koz

Любое равенство регулярных выражений выводимо из аксиоматики **Sal** и аксиоматики **Koz**.

Пример вывода в системе **Koz**:

- (0) $x^* = xx^* \mid \varepsilon = xx^* \mid xx^* \mid \varepsilon = xx^* \mid x^*$ (Unfold + Idm)
- (1) $x \mid yx = x \Rightarrow x \mid y^*x = x$ (Koz, $p = y, q = x$)
- (2) $x^*x^* \mid x^* = x^*$ ($0 + 1$)
- (3) $x^*x^* = (\varepsilon \mid xx^*)(\varepsilon \mid xx^*)$ (Unfold)
- (4) $(\varepsilon \mid xx^*)(\varepsilon \mid xx^*) = (\varepsilon \mid xx^* \mid xx^*) \mid xx^*$ (Dstr + 3)
- (5) $(\varepsilon \mid xx^* \mid xx^*) \mid xx^* = x^* \mid xx^*$ (Idm + Unfold + 4)
- (6) $x^* \mid xx^* = x^* \mid (x^* \mid xx^*)$ (Idm + 5)
- (7) $x^* \mid (x^* \mid xx^*) = x^* \mid x^*x^*$ ($4 + 5$)
- (8) $x^*x^* \leq x^* \ \& \ x^* \leq x^*x^*$ ($2 + 7$)
- (9) $x^*x^* = x^*$ (8)



Смысл леммы Ардена и аксиом Козена

Неподвижная точка функции $f(x)$ — такое x , что $f(x) = x$.

Пусть $X = (pX) \mid q$, где X — неизвестное \mathcal{RE} , а p, q — известные, причём $\varepsilon \notin \mathcal{L}(p)$. Тогда $X = (p)^*q$.

То есть p^*q — наименьшая (но не единственная) неподвижная точка выражения $px \mid q$ по отношению \leq , и единственная, если $\varepsilon \notin \mathcal{L}(p)$.

Рассмотрим систему уравнений:

$$X_1 = (A_{11}X_1) \mid (A_{12}X_2) \mid \dots \mid B_1$$

$$X_2 = (A_{21}X_1) \mid (A_{22}X_2) \mid \dots \mid B_2$$

...

$$X_n = (A_{n1}X_1) \mid (A_{n2}X_2) \mid \dots \mid B_n$$

Положим $\varepsilon \notin A_{ij}$. Выразим X_1 через X_2, \dots, X_n , X_2 через X_3, \dots, X_n и т.д. Получим регулярное выражение для X_n (и после обратных подстановок также для X_{n-1}, \dots, X_1).



От грамматики и НКА к \mathcal{RE}

Теорема Клини

По каждому НКА можно построить \mathcal{RE} , распознающую тот же язык. Верно и обратное.

Здесь считаем, что в НКА нет ε -переходов.

- Объявляем каждый нетерминал (или состояние НКА) переменной и строим для него уравнение:
 - По правилу $A \rightarrow aB$ (или для стрелки из A в B) добавляем альтернативу aB ;
 - По правилу $A \rightarrow b$ (или для стрелки в финальное состояние) добавляем альтернативу без переменных.
 - Если начальное состояние финальное, добавляем в уравнение для S альтернативу ε .
- Решаем систему относительно S .



От грамматики к \mathcal{RE}

Построим \mathcal{RE} по грамматике:

$$\begin{array}{ll} S \rightarrow aT & S \rightarrow aS \\ T \rightarrow aT & T \rightarrow bT \quad T \rightarrow b \end{array}$$

Строим по правилам грамматики систему:

$$S = (aS) \mid (aT)$$

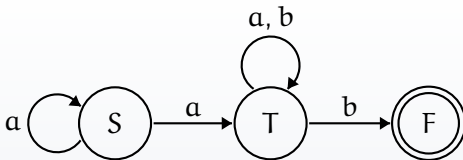
$$T = ((a \mid b)T) \mid b$$

Решаем второе уравнение: $T = (a \mid b)^*b$

Подставляем в первое: $S = (aS) \mid (a(a \mid b)^*b)$ Получаем ответ:

$$S = a^*a(a \mid b)^*b$$

Построим НКА, соответствующий этой грамматике:





От грамматики к \mathcal{RE}

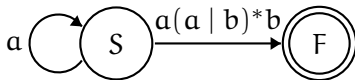
Построим \mathcal{RE} по грамматике:

$$\begin{array}{lll} S \rightarrow aT & S \rightarrow aS \\ T \rightarrow aT & T \rightarrow bT & T \rightarrow b \end{array}$$

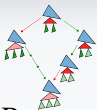
Получаем ответ: $S = a^*a(a \mid b)^*b$

Построим НКА, соответствующий этой грамматике .

Видно, что решив уравнение для T , по существу мы превратили его в НКА над регексами, имеющий на одно состояние меньше.



Можно было бы сказать, что и выражение для $\mathcal{L}(S)$ соответствует НКА с одним переходом (из стартового состояния в финальное), если бы до S было «самое стартовое» состояние с переходом в S по ϵ . Это наблюдение приводит к «двойнику» решения уравнений по лемме Ардена — методу устранения состояний.



От грамматики к \mathcal{RE}

В качестве промежуточной структуры здесь используется НКА с переходами по регулярным выражениям, а не по элементам алфавита.

Метод устранения состояний

- Для единообразия перед преобразованием вводится новое начальное состояние S с ε -переходом в начальное состояние q_0 , и финальное состояние T , с ε -переходами в него из всех $q \in F$. Все состояния, кроме T , становятся нефинальными.
- Пусть требуется устранить состояние q такое, что $q \xrightarrow{\tau} q$. Тогда для всех пар q_A, q_B , где $q_A \xrightarrow{\Phi} q$, $q \xrightarrow{\Psi} q_B$ (q_A и q_B могут совпадать), добавляем переход $q_A \xrightarrow{\Phi(\tau)^*\Psi} q_B$, и после всех таких добавлений удаляем q .
- Когда останутся только S и T , где $S \xrightarrow{P} T$, то P и будет искомым регулярным выражением.



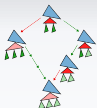
От грамматики к \mathcal{RE}

Построим \mathcal{RE} по грамматике:

$$\begin{array}{ll} S \rightarrow aT & S \rightarrow aS \\ T \rightarrow aT & T \rightarrow bT \quad T \rightarrow b \end{array}$$

Получаем ответ: $S = a^*a(a \mid b)^*b$

Если S выразить через T , получаем язык $\mathcal{L}(S) = a^*a(a \mid b)^*b$.
Точно такое же выражение получится, если сначала применить к S лемму Ардена, а потом подставить туда результат вычисления $\mathcal{L}(T)$. Можно ли гарантировать, что любой порядок подстановок приведёт к одному и тому же результату?



Порядок исключения состояний

Рассмотрим систему частично приведённых правил

$$\begin{array}{lll} S \rightarrow \alpha_1 S & S \rightarrow \beta_1 T & S \rightarrow \gamma_1 \\ T \rightarrow \alpha_2 S & T \rightarrow \beta_2 T & T \rightarrow \gamma_2 \end{array}, \text{ где } \alpha_i, \beta_i, \gamma_i \text{ — регулярные}$$

выражения, включающие в себя, возможно, нетерминалы, отличные от S и T (и не включающие S и T).

По правилу Ардена выразим T через S и подставим результат в S :

$$\begin{aligned} T &= \beta_2^*(\alpha_2 S \mid \gamma_2) \\ S &= \alpha_1 S \mid \beta_1 \beta_2^* \alpha_2 S \mid (\gamma_1 \mid \beta_1 \beta_2^* \gamma_2) \\ S &= (\alpha_1 \mid \beta_1 \beta_2^* \alpha_2)^*(\gamma_1 \mid \beta_1 \beta_2^* \gamma_2) \end{aligned} \quad \text{(TS)}$$

А теперь наоборот:

$$\begin{aligned} S &= \alpha_1^*(\beta_1 T \mid \gamma_1) \\ T &= \beta_2 T \mid \alpha_2 \alpha_1^* \beta_1 T \mid (\gamma_2 \mid \alpha_2 \alpha_1^* \gamma_1) \\ T &= (\beta_2 \mid \alpha_2 \alpha_1^* \beta_1)^*(\gamma_2 \mid \alpha_2 \alpha_1^* \gamma_1) \\ S &= \alpha_1^* \gamma_1 \mid \alpha_1^* \beta_1 (\beta_2 \mid \alpha_2 \alpha_1^* \beta_1)^*(\gamma_2 \mid \alpha_2 \alpha_1^* \gamma_1) \end{aligned} \quad \text{(ST)}$$

Как получить выражение (ST) из (TS) напрямую?

Подчёркивается терм, который будет преобразован на следующем шаге; хайлайтером выделен результат преобразования.

$$\begin{aligned}
 \text{(TS)} \quad & (\underline{\alpha_1} \mid \beta_1 \beta_2^* \alpha_2)^* (\gamma_1 \mid \beta_1 \beta_2^* \gamma_2) \\
 (0) \quad & (\alpha_1^* \beta_1 \beta_2^* \alpha_2)^* \alpha_1^* (\gamma_1 \mid \beta_1 \beta_2^* \gamma_2) \quad (\text{TS} + \text{Dnst}) \\
 (1) \quad & ((\alpha_1^* \beta_1 \beta_2^* \alpha_2)^* \alpha_1^* \gamma_1) \mid ((\underline{\alpha_1^* \beta_1 \beta_2^* \alpha_2})^* \alpha_1^* \underline{\beta_1 \beta_2^* \gamma_2}) \quad (0 + \text{Dstr}) \\
 & \underbrace{\hspace{10em}}_{\Phi} \\
 (2) \quad & \Phi \mid (\alpha_1^* \beta_1 (\underline{\beta_2^* \alpha_2} \alpha_1^* \beta_1)^* \underline{\beta_2^* \gamma_2}) \quad (1 + \text{Sld}) \\
 (3) \quad & ((\underline{\alpha_1^* \beta_1 \beta_2^* \alpha_2})^* \alpha_1^* \gamma_1) \mid (\alpha_1^* \beta_1 (\underline{\beta_2} \mid \alpha_2 \alpha_1^* \beta_1)^* \gamma_2) \quad (2 + \text{Dnst}) \\
 & \underbrace{\hspace{10em}}_{\Psi} \\
 (4) \quad & \alpha_1^* \gamma_1 \mid (\underline{\alpha_1^* \beta_1 \beta_2^* \alpha_2})^* \alpha_1^* \beta_1 \beta_2^* \alpha_2 \alpha_1^* \gamma_1 \mid \Psi \quad (3 + \text{Unfold} + \text{Dstr}) \\
 (5) \quad & \alpha_1^* \gamma_1 \mid \alpha_1^* \beta_1 (\underline{\beta_2^* \alpha_2} \alpha_1^* \beta_1)^* \underline{\beta_2^* \alpha_2} \alpha_1^* \gamma_1 \mid \Psi \quad (4 + \text{Sld}) \\
 (6) \quad & \alpha_1^* \gamma_1 \mid \alpha_1^* \beta_1 (\underline{\beta_2} \mid \alpha_2^* \alpha_1^* \beta_1)^* \alpha_2 \alpha_1^* \gamma_1 \mid \Psi \quad (5 + \text{Dnst}) \\
 \text{(ST)} \quad & \alpha_1^* \gamma_1 \mid \alpha_1^* \beta_1 (\beta_2 \mid \alpha_2^* \alpha_1^* \beta_1)^* (\alpha_2 \alpha_1^* \gamma_1 \mid \gamma_2) \quad (6 + \text{Dstr})
 \end{aligned}$$

Терм (ST) получился из (TS) применением только базовых алгебраических преобразований, развёртки, сдвига и уплощения.



Порядок исключения состояний

Лемма (Conway, Krob)

Регулярные выражения, полученные методом устранения состояний (или применением леммы Ардена), эквивалентны с точностью до преобразований Dstr, Unfold + Sld и Dnst.

Действительно, различные упорядочения состояний по приоритету их исключения могут быть преобразованы друг в друга посредством парных транспозиций. А вывод выражений, полученных транспозицией, мы уже построили. Более того, он симметричный: нигде не используются схемы аксиом, включающие импликацию, т.е. все преобразования можно совершить и в обратную сторону.

С точки зрения практики, выбор порядка исключения может изменить асимптотику скорости сопоставления с выражением.