

# Автомат Глушкова

Лучшая команда разработчиков по ТФЯ

2022 г.

# Линеаризация

## Определение

Если регулярное выражение  $r \in \mathcal{RE}$  содержит  $n$  вхождений букв алфавита  $\Sigma$ , тогда линеаризованное регулярное выражение  $\text{Linearize}(r)$  получается из  $r$  приписыванием  $i$ -ой по счёту букве, входящей в  $r$ , индекса  $i$ .

## Пример

Рассмотрим регулярное выражение:

$$(ba \mid b)aa(a \mid ab)^*$$

Его линеаризованная версия:

$$(b_1a_2 \mid b_3)a_4a_5(a_6 \mid a_7b_8)^*$$

# Множества First, Last, Follow

## Определение

Пусть  $r \in \mathcal{RE}$ , тогда:

- множество First — это множество букв, с которых может начинаться слово из  $\mathcal{L}(r)$  (если  $\varepsilon \in \mathcal{L}(r)$ , то оно формально добавляется в First);
- множество Last — это множество букв, которыми может заканчиваться слово из  $\mathcal{L}(r)$ ;
- множество Follow( $c$ ) — это множество букв, которым может предшествовать  $c$ . Т.е.  $\{d \in \Sigma \mid \exists w_1, w_2 (w_1 c w_2 \in \mathcal{L}(r))\}$ .

## Achtung!

Множество Follow в теории компиляции обычно определяется иначе — это множество символов, которые могут идти за выводом из определённого нетерминального символа. Два этих определения можно унифицировать, если рассматривать каждую букву в  $r$  как «обёрнутую» (в смысле, например, н.ф. Хомского).

# First, Last, Follow — пример

Построим указанные множества для регулярного выражения  
 $r = (ba \mid b)aa(a \mid ab)^*$ .

Начнём с исходного регулярного выражения.

## Исходное регулярное выражение

- $\text{First}(r) = \{b\}$ .
- $\text{Last}(r) = \{a, b\}$ .
- $\text{Follow}_r(a) = \{a, b\}$ ;  $\text{Follow}_r(b) = \{a\}$ .

Хотя данные множества описывают, как устроены слова из  $\mathcal{L}(r)$  локально, однако они не исчерпывают всей информации о языке, поскольку разные вхождения букв в регулярное выражения никак не различаются.

Например, по множествам First и Last можно предположить, что  $b \in \mathcal{L}(r)$ , хотя это не так.

# First, Last, Follow — пример

Построим указанные множества для регулярного выражения  
 $r = (ba \mid b)aa(a \mid ab)^*$ .

Вспомним, что  $r_{Lin} = (b_1a_2 \mid b_3)a_4a_5(a_6 \mid a_7b_8)^*$ .

## Линеаризованное выражение

- $\text{First}(r_{Lin}) = \{b_1, b_3\}$ .
- $\text{Last}(r_{Lin}) = \{a_5, a_6, b_8\}$ .
- $\text{Follow}_{r_{Lin}}(b_1) = \{a_2\}$ ;  $\text{Follow}_{r_{Lin}}(a_2) = \{a_4\}$ ;  $\text{Follow}_{r_{Lin}}(b_3) = \{a_4\}$ ;  
 $\text{Follow}_{r_{Lin}}(a_4) = \{a_5\}$ ;  $\text{Follow}_{r_{Lin}}(a_5) = \{a_6, a_7\}$ ;  
 $\text{Follow}_{r_{Lin}}(a_6) = \{a_6, a_7\}$ ;  $\text{Follow}_{r_{Lin}}(a_7) = \{b_8\}$ ;  
 $\text{Follow}_{r_{Lin}}(b_8) = \{a_6, a_7\}$ .

В описании данных множеств содержится исчерпывающая информация о языке  $\mathcal{L}(r_{Lin})$ .

# Конструкция автомата Глушкова

## Алгоритм построения $\text{Glushkov}(r)$

- Строим линеаризованную версию  $r$ :  $r_{\text{Lin}} = \text{Linearize}(r)$ .
- Находим  $\text{First}(r_{\text{Lin}})$ ,  $\text{Last}(r_{\text{Lin}})$ , а также  $\text{Follow}_{r_{\text{Lin}}}(c)$  для всех  $c \in \Sigma_{r_{\text{Lin}}}$ .
- Все состояния автомата, кроме начального (назовём его  $S$ ), соответствуют буквам  $c \in \Sigma_{r_{\text{Lin}}}$ .
- Из начального состояния строим переходы в те состояния, для которых  $c \in \text{First}(r_{\text{Lin}})$ . Переходы имеют вид  $S \xrightarrow{c}$ .
- Переходы из состояния  $c$  соответствуют элементам  $d$  множества  $\text{Follow}_{r_{\text{Lin}}}(c)$  и имеют вид  $c \xrightarrow{d}$ .
- Конечные состояния — такие, что  $c \in \text{Last}(r_{\text{Lin}})$ , а также  $S$ , если  $\varepsilon \in \mathcal{L}(R)$ .
- Теперь стираем разметку, построенную линеаризацией, на переходах автомата. Конструкция завершена.

# Пример автомата Глушкова

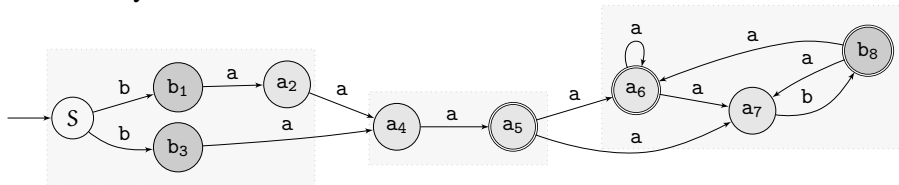
Исходное регулярное выражение:

$$(ba \mid b)aa(a \mid ab)^*$$

Линеаризованное регулярное выражение:

$$(b_1a_2 \mid b_3)a_4a_5(a_6 \mid a_7b_8)^*$$

Автомат Глушкова:



Подграфы, распознающие регулярные выражения, являющиеся подструктурами исходного, не имеют общих вершин. Это свойство автомата Глушкова используется в реализациях match-функций некоторых библиотек регулярных выражений.

# Свойства автомата Глушкова

- Не содержит  $\varepsilon$ -переходов.
- Число состояний равно длине регулярного выражения (без учёта регулярных операций), плюс один (стартовое состояние).
- В общем случае недетерминированный.

## Примечание

Для 1-однозначных регулярных выражений  $r$  автомат  $\text{Glushkov}(r)$  является детерминированным. Эту его особенность активно используют в современных библиотеках регулярных выражений, например, в RE2. Выигрыш может получиться колоссальным: например,  $\text{Thompson}((a^*)^*)$  является экспоненциально неоднозначным, а  $\text{Glushkov}((a^*)^*)$  однозначен и детерминирован!