Теория автоматов

Лекция 3: Регулярные языки

Дьулустан Никифоров

Кафедра ИТ
Северо-Восточный Федеральный Университет

Осень 2024

Конечные Автоматы

Definition

Пусть $M = (Q, \Sigma, \delta, S, F)$ - конечный автомат, $w = w_1 w_2 \dots w_n$ — строка над Σ.
Тогда мы говорим, что M **принимает (accepts)** w, если есть последовательность $r_0, r_1, \dots, r_n$ in Q такая, что:

- $r_0 = S$,
- $\delta(r_i, w_{i+1}) = r_{i+1}$ for $i = 0, \dots, n-1$, and
- $r_n \in F$.

Конечные Автоматы

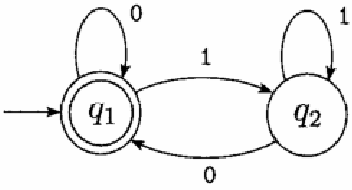
- Если A — это множество всех строк, *принимаемых* машиной M, то мы говорим, что **M распознает/принимает (recognizes/accepts) A**.
- **A — язык (language)** машины **M**, обозначается **A=L(M)**.
- Обратите внимание, всегда можно сказать, что M принимает пустую строку.

Definition

Язык называется **регулярным языком (regular language)**, если какой-то конечный автомат распознает его.

Конечные Автоматы: Примеры

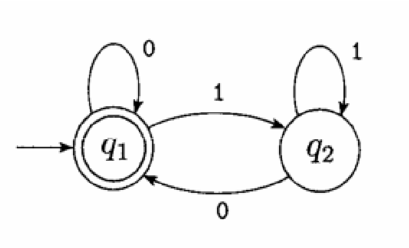
Нам дан такой автомат M_1:



Какой язык распознается этим автоматом?

Конечные Автоматы: Примеры

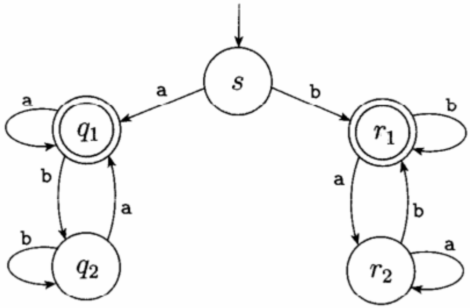
Нам дан такой автомат M_1:



Какой язык распознается этим автоматом?

$L(M_1) = \{w \mid w$ пустая строка или заканчивается на $0\}$.

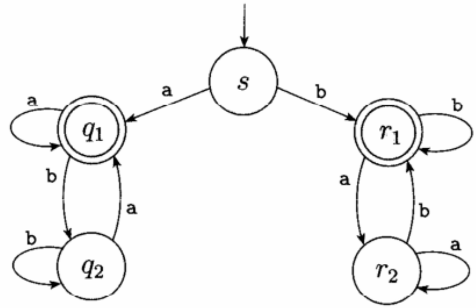
Нам дан такой автомат M_2:



Какой язык распознается этим автоматом?

Конечные Автоматы: Примеры

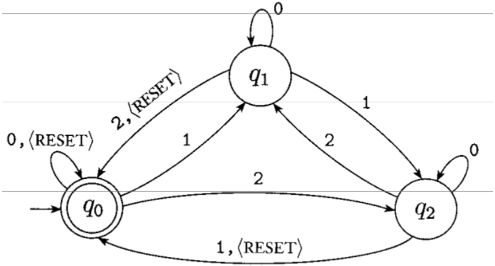
Нам дан такой автомат M_2:



Какой язык распознается этим автоматом?

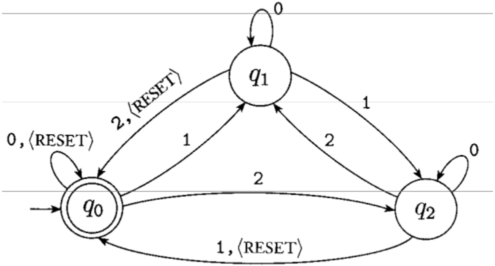
$L(M_2) =$
$\{w \mid w$ начинается и заканчивается на одинаковые символы$\}$.

Нам дан такой автомат M_3:



Какой язык распознается этим автоматом?

Нам дан такой автомат M_3:



Какой язык распознается этим автоматом?
$\Sigma = \{0, 1, 2, \langle RESET \rangle\}$,
Автомат суммирует получаемые числа по модулю 3 (т.е. поддерживает остаток от деления суммы чисел на 3), но если получает символ $\langle RESET \rangle$, то обнуляет сумму. Принимает строку, если в конце сумма равна 0.

Конечные Автоматы

- В задачах на автоматы, вам часто надо будет самим придумывать автоматы такие, что они распознают заданный язык. Как надо подходить к таким задачам?
- Это креативный процесс, и нельзя дать точный алгоритм процессу решения таких задач. Но помогает следовать принципу:
  представьте себе, что вы — это автомат.
- Когда вы последовательно читаете строку, подумайте, какую информацию следует запомнить, а какую можно забыть?
- Следите, какой информацией обладает автомат на каждом своем состоянии! Состояние — это единственный источник информации для автомата.

Регулярные языки

Вернемся к теории.

- По определению, регулярные языки — это языки, распознаваемые конечными автоматами.
- Значит, множество регулярных языков — это множество всех "задач", которые решимы при помощи простых машин, называемых конечными автоматами.
- Мы хотим изучить, как устроено это множество.

Регулярные операции

Пусть A и B языки. Тогда мы определяем такие **регулярные операции (regular operations)**:

- **Объединение (Union)** A и B:
  $A \cup B = \{x \mid x \in A \text{ или } x \in B\}$ — строки, которые принадлежат хотя бы одному из A и B.
- **Конкатенация (Concatenation)** A и B:
  $AB = \{xy \mid x \in A \text{ и } x \in B\}$.
- **Замыкние/“звездочка” (Closure/Star)**:
  $A^* = \{x_1 x_2 \dots x_k \mid k \geq 0 \text{ и каждый } x_i \in A\}$.

Заметьте, что всегда $\varepsilon \in A^*$.

Регулярные операции: Пример

Пусть Σ — латинский алфавит, и нам даны языки $A = \{good, bad\}, B = \{cop, criminal\}$.

Регулярные операции: Пример

Пусть Σ — латинский алфавит, и нам даны языки $A = \{good, bad\}, B = \{cop, criminal\}$.

- $A \cup B = \{good, bad, cop, criminal\}$
- $AB = \{goodcop, badcop, goodcriminal, badcriminal\}$
- $A^* = \{\varepsilon, good, bad, goodgood, goodbad, badgood, badbad, goodgoodgood, goodgoodbad, \dots\}$
- $B^* = \{\varepsilon, cop, criminal, copcop, copcriminal, \dots\}$

Регулярные языки: Объединение

Theorem

Множество регулярных языков замкнуто под операцией объединения, т.е.
A_1 *и* A_2 *— регулярные языки* $\Rightarrow A_1 \cup A_2$ *— регулярный язык.*

Proof:

Регулярные языки: Объединение

Theorem

Множество регулярных языков замкнуто под операцией объединения, т.е.
A_1 *и* A_2 *— регулярные языки* $\Rightarrow A_1 \cup A_2$ *— регулярный язык.*

Proof:
Мы знаем, что существуют конечные автоматы M_1 и M_2 т.ч. M_1 распознает A_1 и M_2 распознает A_2. Нам надо доказать, что существует конечный автомат M, который распознает $A_1 \cup A_2$.

Proof:
Мы построим такой автомат M, который будет одновременно симулировать и машину M_1, и машину M_2.
$M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$
$M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$
Какой сделаем M?

Proof:

- $Q = Q_1 \times Q_2$ — создаем состояния как всевозможные пары состояний между M_1 и M_2;
- $\delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a))$ — переходы одновременно делаем согласно переходам обоих машин;
- $q_0 = (q_1, q_2)$ — начальное состояние соответствует комбинации начальных состоний машин;
- $F = \{(r_1, r_2) \mid r_1 \in F_1 \text{ or } r_2 \in F_2\}$ — для принятия строки достаточно, чтобы приняла хотя бы одна машина.

Можно видеть, что такая машина принимает строку тогда и только тогда, когда машина M_1 или машина M_2 принимает ее.

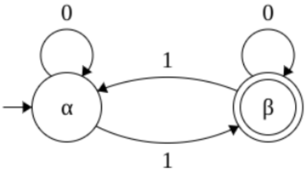
Регулярные языки: Объединение

Сделаем такое построение на конкретном примере:

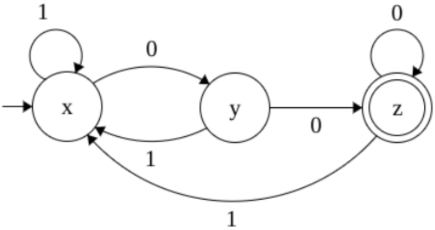
- $\Sigma = \{0, 1\}$;
- A_1 — строки с нечетным кол-вом единиц;
- A_2 — строки, оканчивающиеся на 00;

Регулярные языки: Объединение

M_1 для A_1:

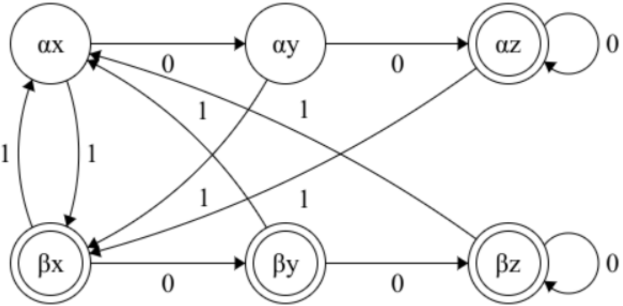


M_2 для A_2:



Регулярные языки: Объединение

M для $A = A_1 \cup A_2$:



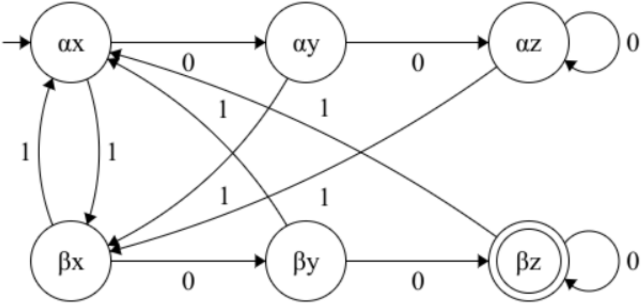
Давайте ещё рассмотрим такие по-факту регулярные операции:

- **Пересечение (intersection)** A и B:
  $A \cap B = \{x \mid x \in A \text{ и } x \in B\}$ — строки, которые принадлежат и A, и B.
- **Разница (difference)** A и B: $A \setminus B = \{x \mid x \in A \text{ и } x \notin B\}$ — строки, которые принадлежат A, но не принадлежат B.
- **Дополнение (complement)** A (с алфавитом Σ):
  $\overline{A} = \Sigma^* \setminus A$ — строки, которые *не* принадлежат языку A.

$A \cap B$, $A \setminus B$ — мы строим автомат ровно так же, как и строили $A \cup B$. Меняется только, какие состояния являются принимающими.

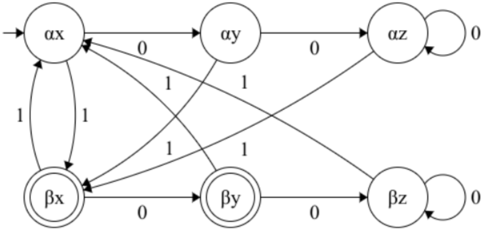
Регулярные языки: Пересечение

M для $A_1 \cap A_2$:

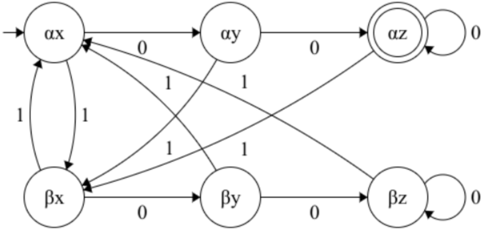


Регулярные языки: Разница

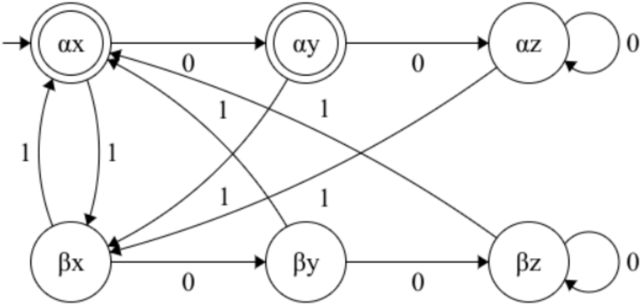
M для $A_1 \setminus A_2$:



M для $A_2 \setminus A_1$:



M для $\overline{A_1 \cup A_2}$:



Регулярные языки: еще пример

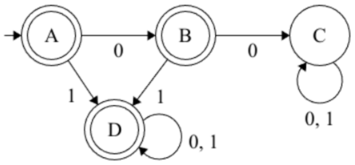
Задача: Постройте конечный автомат, который принимает язык бинарных строк, не начинающихся на 00 И заканчивающихся на 01.

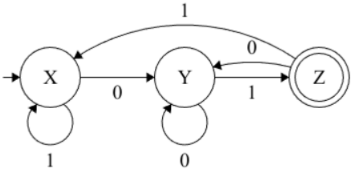
Решение 1: Построим отдельно конечные автоматы для языка $L_1 = \{w \in \{0,1\}^* \mid w$ не начинается на $00\}$ и для языка $L_2 = \{w \in \{0,1\}^* \mid w$ заканчивается на $01\}$

Регулярные языки: еще пример

Задача: Постройте конечный автомат, который принимает язык бинарных строк, не начинающихся на 00 И заканчивающихся на 01.

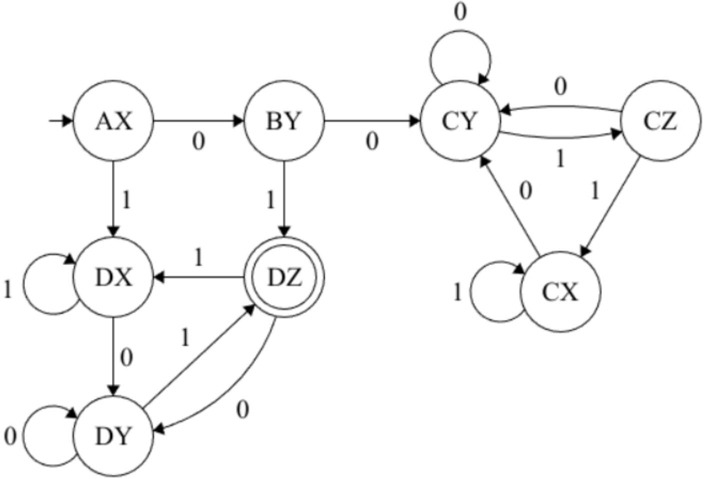
Решение 1: Построим отдельно конечные автоматы для языка $L_1 = \{w \in \{0,1\}^* \mid w$ не начинается на 00$\}$ и для языка $L_2 = \{w \in \{0,1\}^* \mid w$ заканчивается на 01$\}$





Регулярные языки: еще пример

Теперь строим конечный автомат для языка $L_1 \cap L_2$ по алгоритму:

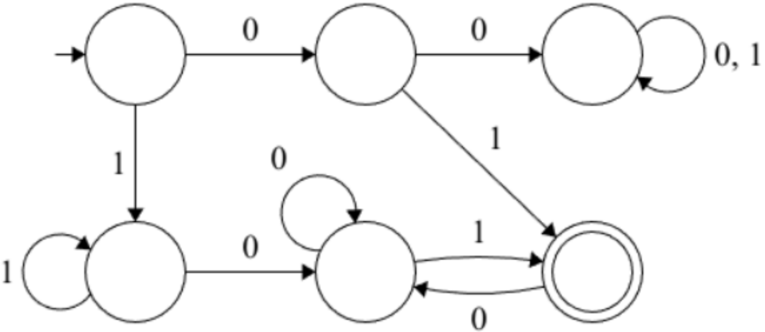


Регулярные языки: еще пример

Решение 2: Сразу построим конечный автомат, используя брейн:

Регулярные языки: еще пример

Решение 2: Сразу построим конечный автомат, используя брейн:



Регулярные языки: Конкатенация

Theorem

Множество регулярных языков замкнуто под операцией конкатенации, т.е.
A_1 *и* A_2 *— регулярные языки* $\Rightarrow A_1A_2$ *— регулярный язык.*

Proof:

Попробуем также построить M, который симулирует действия автоматов M_1 и M_2. Но теперь все сложнее! Почему?

Регулярные языки: Конкатенация

Theorem

Множество регулярных языков замкнуто под операцией конкатенации, т.е.
A_1 *и* A_2 *— регулярные языки* $\Rightarrow A_1 A_2$ *— регулярный язык.*

Proof:

Попробуем также построить M, который симулирует действия автоматов M_1 и M_2. Но теперь все сложнее! Почему?

M должна сначала принять какую-то часть строки согласно машине M_1, а потом принять оставшуюся часть строки согласно M_2. Но где провести такую границу? Непонятно...

Недетерминированные Конечные Автоматы

Чтобы справиться с такой проблемой, нам надо ввести новое понятие — nondeterminism (недетерменированность).

- Конечные автоматы, которые были у нас до этого — называются **Deterministic Finite Automata (DFA)** / Детерминированные конечные автоматы (ДКА).
- Введем новое определение конечных автоматов — **Nondeterministic Finite Automata (NFA)** / Недетерминированные конечные автоматы (НКА).
- Теперь мы не требуем, чтобы для каждого состояния для каждого символа была ровно одна стрелка. Стрелок может быть 0, 1 или больше.
- Также у нас есть стрелки, обозначенные через ε — это стрелки, которые «бесплатно» переходят в другое состояние.