Теория автоматов

Лекция 2: Формальные языки и конечные автоматы

Дьулустан Никифоров

Кафедра ИТ
Северо-Восточный Федеральный Университет

Осень 2024

Формальные языки (Formal languages)

- Принимаем какое-нибудь *конечное* множество за алфавит: Σ — **алфавит** (**alphabet**).
- Примеры: $\Sigma_1 = \{0, 1\}$, $\Sigma_2 = \{a, b, \ldots, z\}$, $\Sigma_3 = \{0, 1, a, b, c\}$, $\Sigma_4 = \{A, B, ..., Z\}$.
  Элементы алфавита будем называть буквами/символами (**letters/symbols**).
- **строка** над алфавитом Σ — *конечная* последовательность символов из Σ.
  string over alphabet Σ — *finite* sequence of symbols in Σ.
- Примеры:
  - $\Sigma = \{a, b, c\} : a, bc, aaabbaac, bbccaaabacabac$;
  - $\Sigma = \{0, 1\} : 0, 1, 00, 101010001$.

Формальные языки

- $|w|$ обозначает длину строки w.
  $|bloodymary| = 10$.
- Строка длины 0 — это пустая(**empty**) строка.
  Пустая строка обозначается ε.

Формальные языки

- Часто пишем $w = a_1 a_2 \dots a_n$ для строки длины n.
- **reverse** of $w = a_1 a_2 \dots a_n$:
  $w^{\mathcal{R}} = a_n a_{n-1} \dots a_1$.
- z — **подстрока (substring)** w если оно содержится внутри w. При этом подстрока может быть пустой строкой или совпадать полностью со всей строкой.
- Примеры:
  gmo — подстрока $bigmom$;
  $devilfruit$ — подстрока $devilfruit$;
  ε — подстрока вообще любой строки.
  $maari$ — *не* является подстрокой $marine$.

Формальные языки

- **конкатенация (concatenation)** строк x и y: приклеиваем y к хвосту x.
  $x = a_1 a_2 \dots a_n, y = b_1 b_2 \dots b_m \Rightarrow xy = a_1 a_2 \dots a_n b_1 b_2 \dots b_m$.
- Часто пишем $x^k = xx \dots x$ (k раз).
  $x^0 = \varepsilon$.

Формальные языки

- Σ^* — множество всех строк над Σ.
  $\Sigma^* = \{w \mid w \text{ строка над } \Sigma\}$.
  $\Sigma^* = \Sigma^0 \bigcup \Sigma^1 \bigcup \Sigma^2 \bigcup \ldots$
- Σ^+ — множество всех непустых строк над Σ.
- **Язык (language)** L над алфавитом Σ — это множество строк над Σ.

Формальные языки

- Σ^* — множество всех строк над Σ.
  $\Sigma^* = \{w \mid w \text{ строка над } \Sigma\}$.
  $\Sigma^* = \Sigma^0 \bigcup \Sigma^1 \bigcup \Sigma^2 \bigcup \ldots$
- Σ^+ — множество всех непустых строк над Σ.
- **Язык (language)** L над алфавитом Σ — это множество строк над Σ.
  $L \subseteq \Sigma^*$.
- Для алфавита $\Sigma = \{0, 1\}$ какие языки можете придумать?

Формальные языки

- Σ^* — множество всех строк над Σ.
  $\Sigma^* = \{w \mid w \text{ строка над } \Sigma\}$.
  $\Sigma^* = \Sigma^0 \bigcup \Sigma^1 \bigcup \Sigma^2 \bigcup \ldots$
- Σ^+ — множество всех непустых строк над Σ.
- **Язык (language)** L над алфавитом Σ — это множество строк над Σ.
  $L \subseteq \Sigma^*$.
- Для алфавита $\Sigma = \{0, 1\}$ какие языки можете придумать?
  $L_{even} = \{w \mid w \in \Sigma^* \text{ и } |w| \text{ четное число }\}$;
  $L = \{0^n 1^n \mid n \in \mathbb{N}\}$;
  $L =$
  $\{w \mid w \in \Sigma^* \text{ и количество 1-ек больше, чем количество 0-ов}\}$.

Формальные языки

- Будем говорить, что “программа” **принимает (accepts)** язык L , если для любой строки $s \in L$ программа возвращает “YES”, а для любой строки $s \notin L$ программа возвращает “NO”.
- В качестве “программы” у нас будут выступать разные вещи — абстрактные машины в течение курса.
- Какие прикольные языки может быть интересно исследовать?

Формальные языки

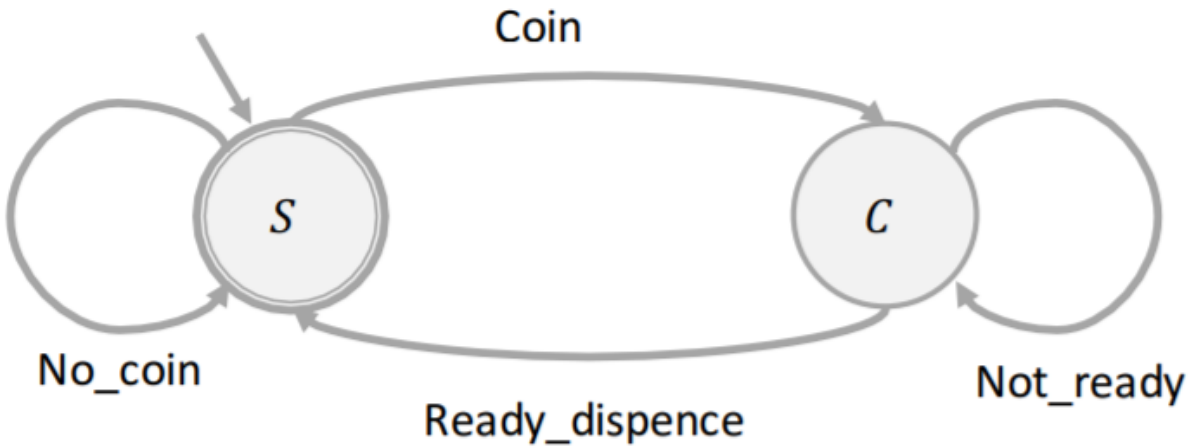
- Будем говорить, что “программа” **принимает (accepts)** язык L , если для любой строки $s \in L$ программа возвращает “YES”, а для любой строки $s \notin L$ программа возвращает “NO”.
- В качестве “программы” у нас будут выступать разные вещи — абстрактные машины в течение курса.
- Какие прикольные языки может быть интересно исследовать?
  - Язык всех простых чисел;
  - Язык всех отсортированных массивов;
  - Язык всех правдивых теорем в данной теории;
  - Многие волнующие вопросы человечества можно поставить таким способом!

Формальные языки

- **Задача (Problem)** — это вопрос определения, принадлежит ли данная строка данному языку.
- Как видите, на самом деле, определить *язык* — это и есть поставить *задачу*.
- Решить задачу — это придумать *“программу”*.

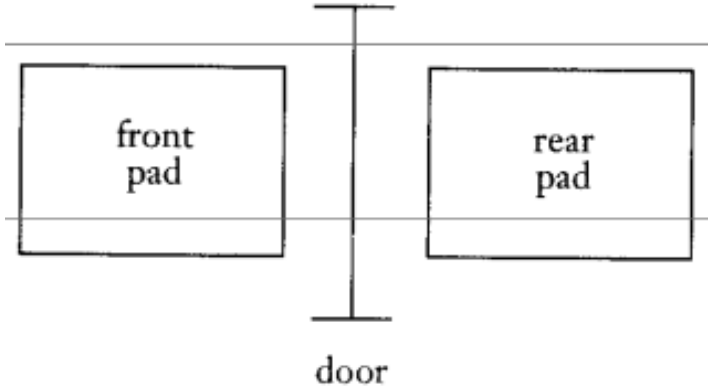
Конечные Автоматы: введение

- Пример конечного автомата: кофе-автомат



Конечные Автоматы: введение

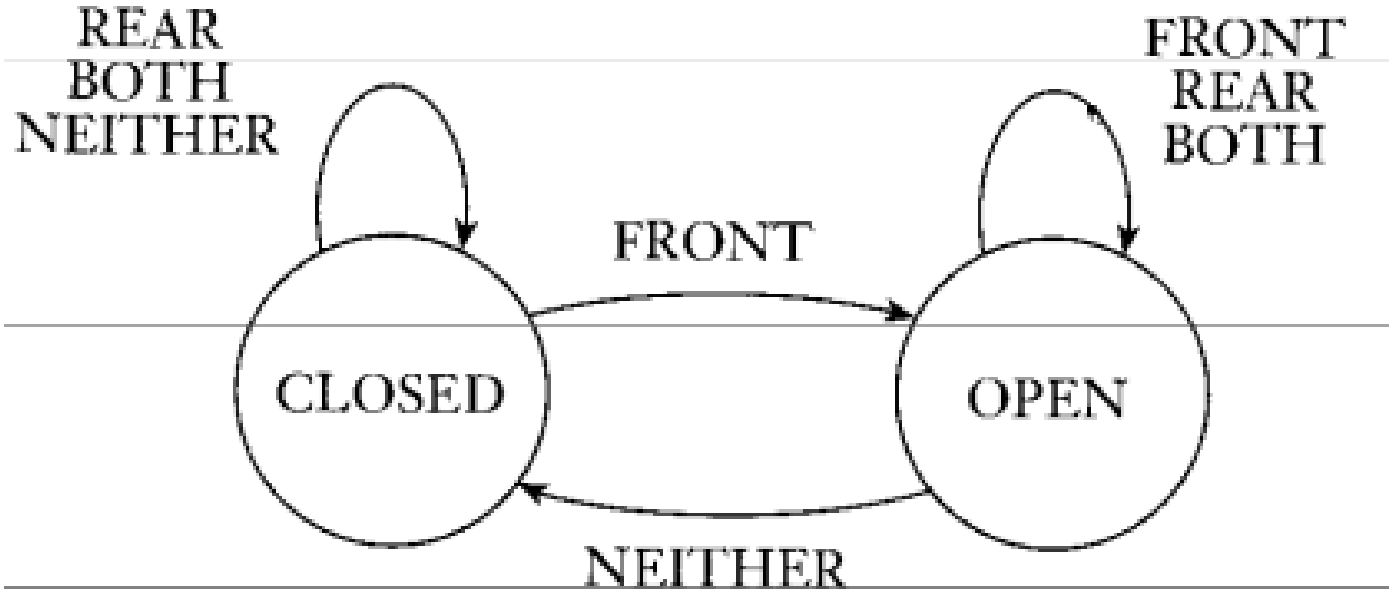
- Пример конечного автомата: автоматическая дверь



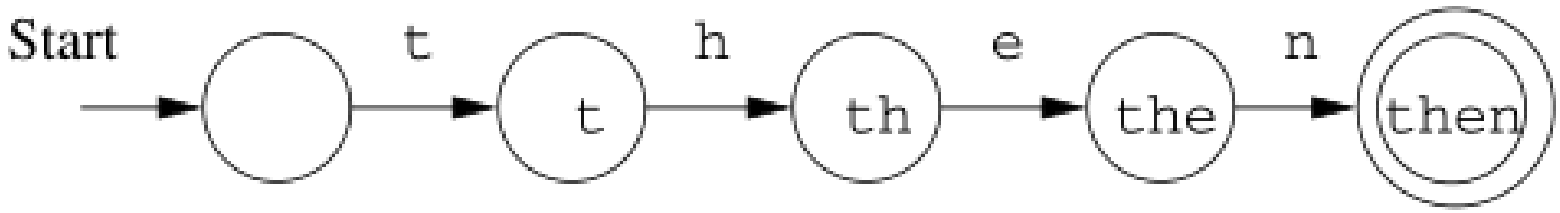
Дверь имеет 2 возможных состояния: открытое (OPEN) и закрытое (CLOSED).
Есть 4 возможных действия:

- FRONT — человек стоит перед дверью;
- REAR — человек стоит сзади двери;
- BOTH — люди стоят перед и сзади двери;
- NEITHER — рядом никого нет;

Конечные Автоматы: введение

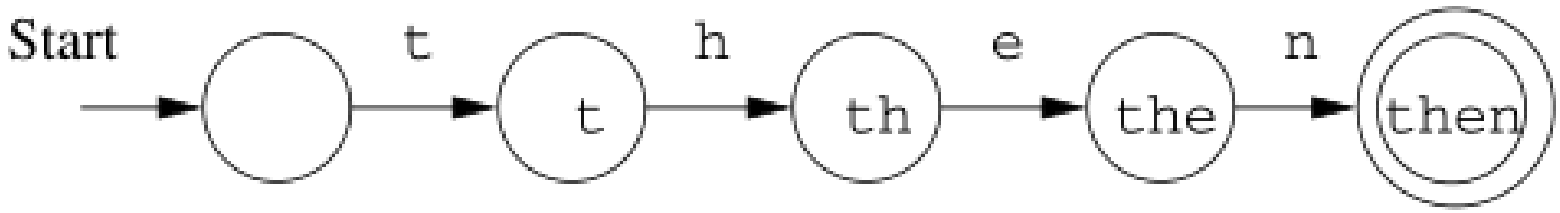


Конечные Автоматы: введение



А это что за автомат?

Конечные Автоматы: введение



А это что за автомат?

Программа, которая ищет слово “then” в тексте.

(!) Однако, это не Детерминированный конечный автомат (ДКА), которые мы сначала будет изучать.

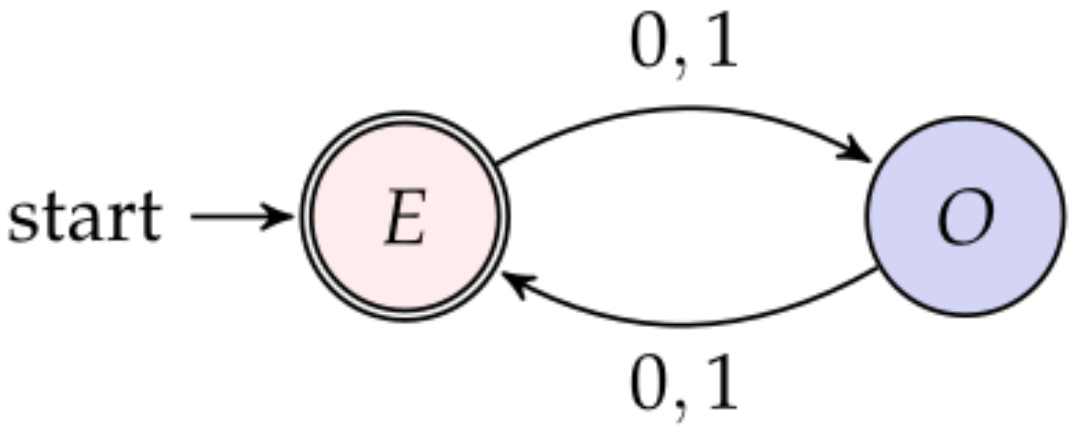
Конечные Автоматы: ментальный процесс

Такие теоретические конструкции могут показаться примитивными/ненужными, но они на удивление хорошо моделируют процесс мышления/вычисления для некоторых задач!

- Распознать строку *четной длины*.
- Распознать строку, содержащую *четное количество 0-ов*.
- Распознать строку, содержащую *нечетное количество 1-ек*.
- Распознать бинарное число, которое *четное*.

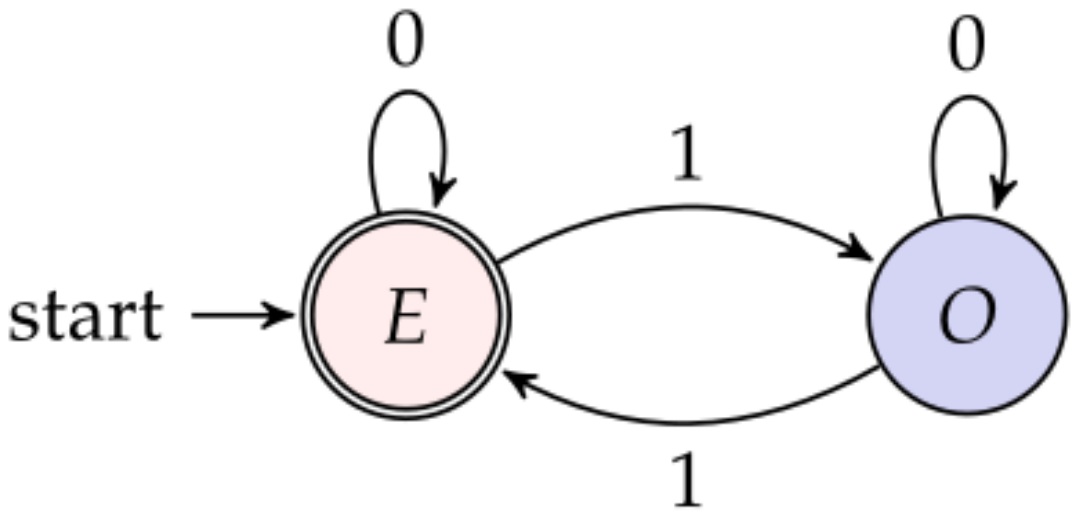
Конечные Автоматы: ментальный процесс

Распознать бинарную строку *четной длины*.



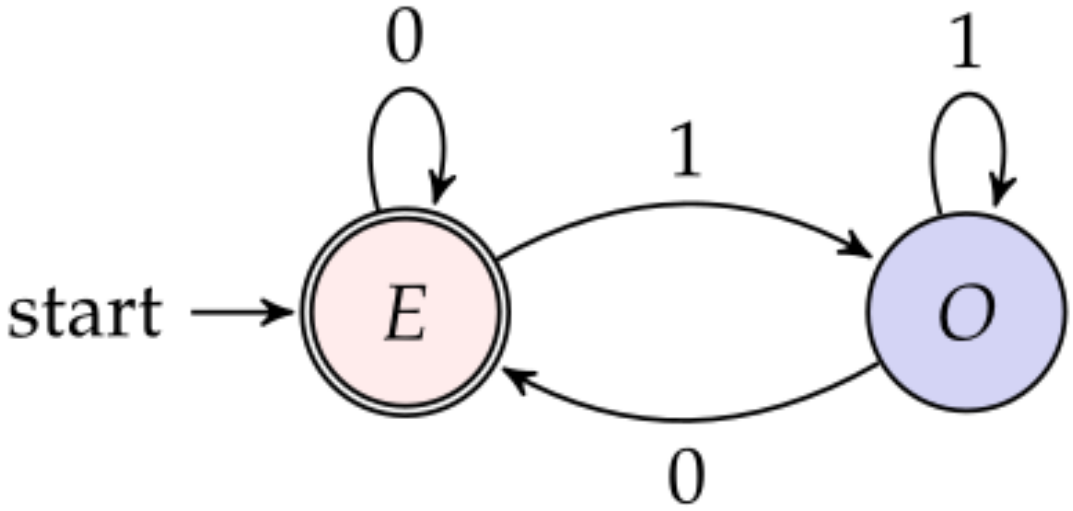
Конечные Автоматы: ментальный процесс

Распознать бинарную строку, содержащую *четное количество 1-ек*.



Конечные Автоматы: ментальный процесс

Распознать бинарное число, которое *четное*.



Конечные Автоматы

- Конечные автоматы — это простая модель компьютеров.
- Вопрос: что мы можем сделать, имея на руках ограниченное кол-во ресурсов?
- Это дает инсайты на что могут делать компьютеры с очень ограниченной памятью.

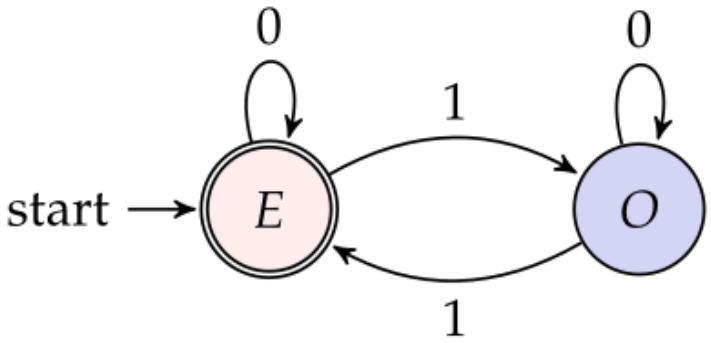
Конечные Автоматы: определение

Definition

Конечный автомат (finite automaton) — это 5-tuple $(Q, \Sigma, \delta, S, F)$, где

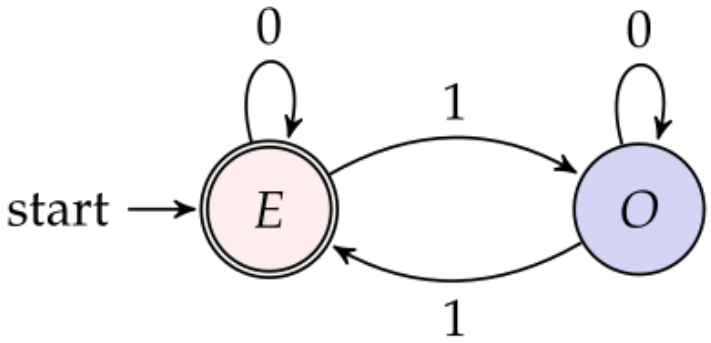
- Q — конечное множество, называемое множеством **состояний (states)**,
- Σ — конечное множество, называемое **алфавитом (alphabet)**,
- $\delta : Q \times \Sigma \to Q$ — **функция перехода (transition function)**,
- $S \in Q$ — **начальное состояние (start state)**,
- $F \subseteq Q$ — **множество принимающих состояний (set of accept states)**.

Конечные Автоматы: формальное описание автомата



Формально опишем этот конечный автомат:

Конечные Автоматы: формальное описание автомата



Формально опишем этот конечный автомат:

- $Q = \{E, O\}$,
- $\Sigma = \{0, 1\}$,
- $S = E$,
- $F = \{E\}$,
- $\delta(E, 0) = E, \delta(E, 1) = O$,
  $\delta(O, 0) = O, \delta(O, 1) = E$.

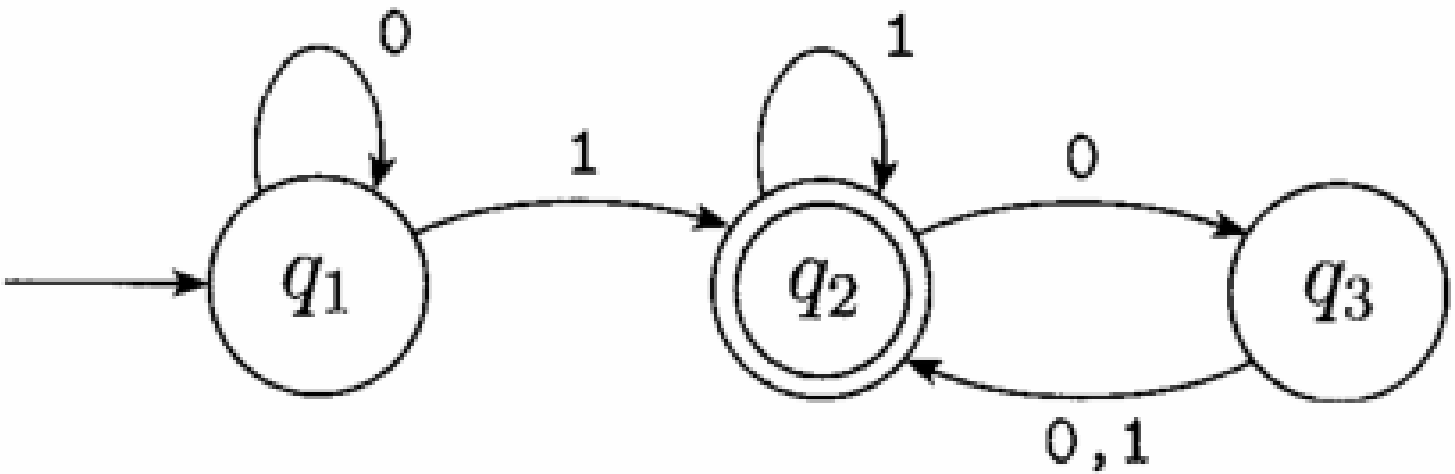
Конечные Автоматы: формальное описание автомата

- $Q = \{q_1, q_2, q_3\}$,
- $\Sigma = \{0, 1\}$,
- $S = q_1$,
- $F = \{q_2\}$,
- δ описано через **таблицу переходов**:

	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

Давайте нарисуем этот конечный автомат.

Конечные Автоматы



Конечные Автоматы: вычисление автомата

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:

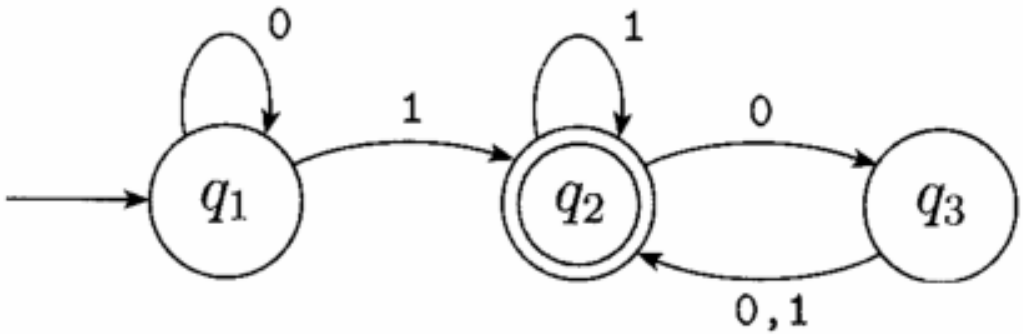
- $r_0 = S$,
- $\delta(r_i, w_{i+1}) = r_{i+1}$ for $i = 0, \dots, n-1$, and
- $r_n \in F$.

Конечные Автоматы: язык автомата

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

Конечные Автоматы: язык автомата

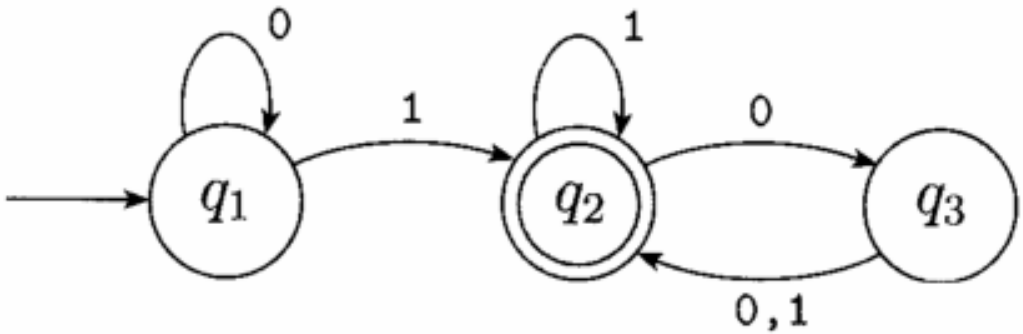
Посмотрим на предыдущий пример:



Построим вычисление этого автомата на строке 00111010000:

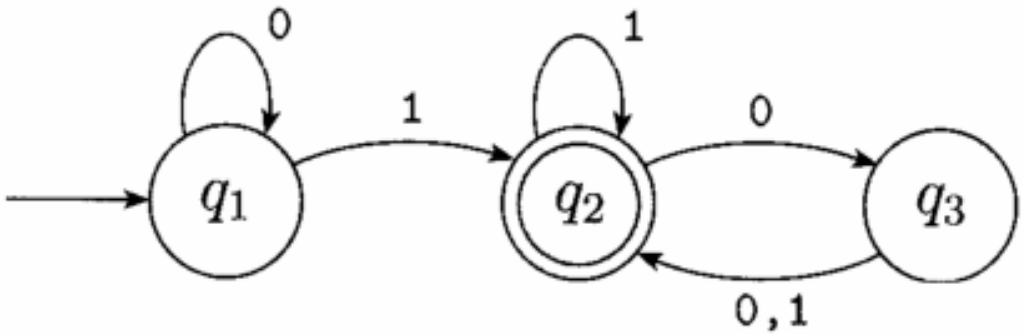
Конечные Автоматы: язык автомата

Посмотрим на предыдущий пример:



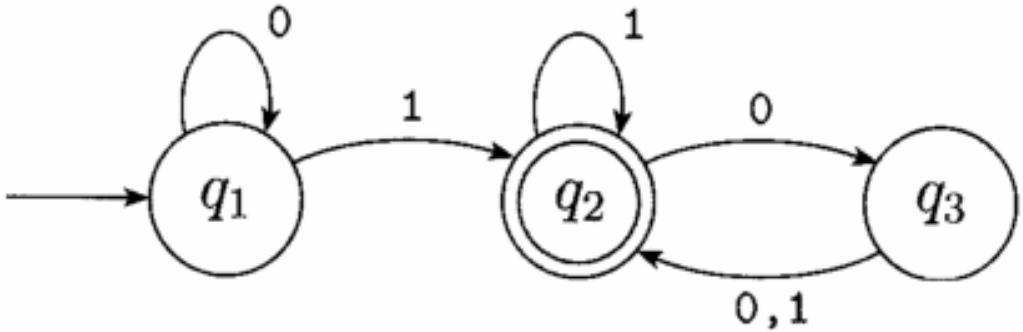
Построим вычисление этого автомата на строке 00111010000:
$q_1 \xrightarrow{0} q_1 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_2 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0} q_2 \xrightarrow{0} q_3 \xrightarrow{0} q_2$.
Автомат принимает строку, так как последнее состояние вычисления — это q_2, являющийся принимающим.

Конечные Автоматы: язык автомата



Что за язык распознается автоматом?

Конечные Автоматы: язык автомата



Что за язык распознается автоматом?

$L(M) = \{w \mid w$ содержит хотя бы одну 1-ку и
четное количество 0-ков следуют за последней 1-кой$\}$.