CSCE 355
1/10/2024

Def: A *deterministic finite automaton* (DFA) is a 5-tuple

$\langle Q, \Sigma, \delta, q_0, F \rangle$ where

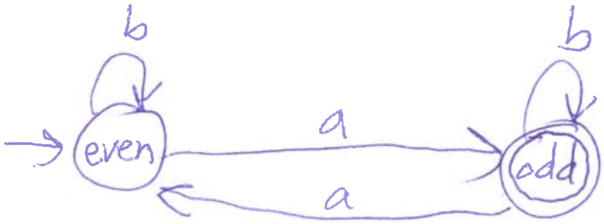
- Q is ~~any~~ a finite set (the *state set*; elements of Q are *states*)
- Σ is an alphabet (the *input alphabet*; all inputs to the DFA are strings over Σ)
- δ (later)
- $q_0 \in Q$ (the *start state*)
- $F \subseteq Q$ (the elements of F are called the *accepting states*; the ~~elements~~ states not in F (in $Q \setminus F$) are the *rejecting states*)
- $\delta : Q \times \Sigma \to Q$

$$q \xrightarrow{a} r \qquad a \in \Sigma$$

means $\delta(q, a) = r$

(the transition function)

Ex: A:



$\Sigma = \{a, b\}$

◎ — accepting state

○ — rejecting state

$A = \langle \{even, odd\}, \{a, b\}, \delta, even, \{odd\} \rangle$

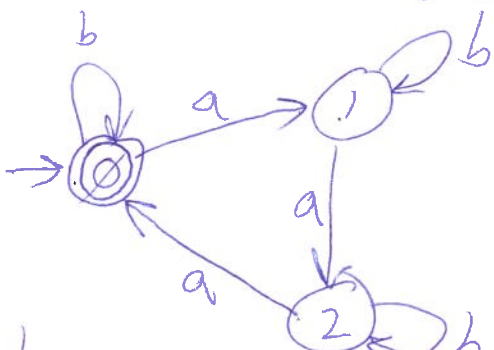
where δ is given by a table:

	a	b
even	odd	even
odd	even	odd

Tabular form of A

	a	b
→ even	odd	even
* odd	even	odd

Ex:



	a	b
→ * ∅	1	∅
1	2	1
2	∅	2

What a DFA does

Def: Let $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ be a DFA, and let $x = x_1 x_2 \cdots x_n$ be a string over Σ ($n \geq 0$) (each $x_i \in \Sigma$). The (computational) trace of A on input x is the unique sequence of states

$s_0, s_1, \ldots, s_n$ such that

1. $s_0 = q_0$ (start state)
2. For all $i \in \{1, \ldots, n\}$,

$$s_i = \delta(s_{i-1}, x_i)$$

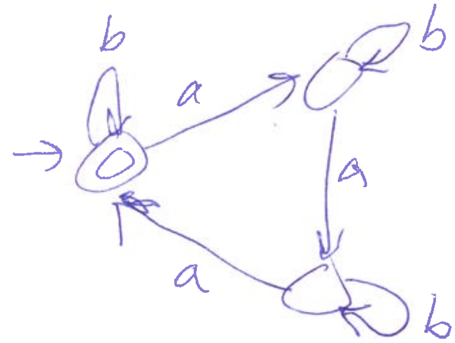
We say that A ends in state s_n on input x. Say that A accepts x if A ends in an accepting state on input x. Otherwise A rejects x (ends in a rejecting state).

Def: Given alphabet Σ, we let Σ^* denote the set of all strings over Σ. A language over Σ is any subset of Σ^* (any set of strings over Σ)

Def: Given a DFA A with input alphabet Σ, The language of A (or the lang. recognized by A) is defined as

$$L(A) := \{x \in \Sigma^* : A \text{ accepts } x\}$$

Ex: $A =$

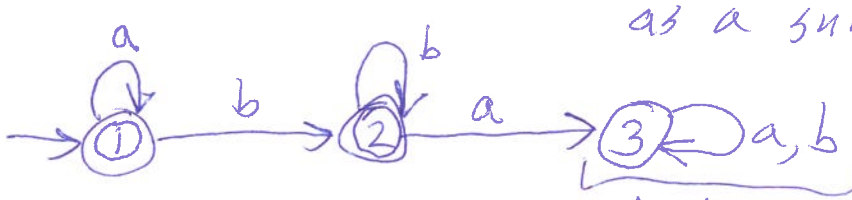


$L(A) = \{x \in \{a,b\}^* :$ ~~x has a~~ the # of a's in x is a multiple of 3$\}$

Def: Let $L \subseteq \Sigma^*$ be a language over alphabet Σ. Say that L is regular if some DFA recognizes L. ($L = L(A)$ for some DFA A.)

Ex: A DFA A that recognizes

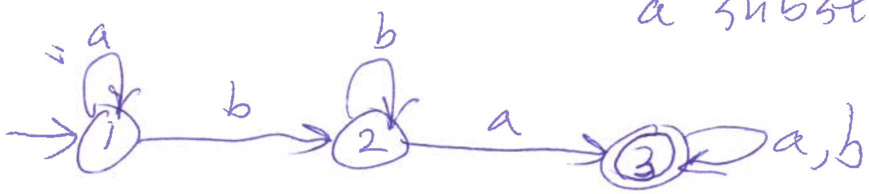
$\{x \in \{a,b\}^* : x$ does not contain ba as a substring$\}$



"dead state"

DFA B that recognizes

$$\{x \in \{a,b\}^* : x \text{ does have } ba \text{ as a substring}\}$$



Def: Let $L \subseteq \Sigma^*$ be a language over Σ. The complement of L (in Σ^*) is the language

$$\overline{L} := \{x \in \Sigma^* : x \notin L\} \quad \left(= \Sigma^* - L = \Sigma^* \setminus L\right)$$

Def: Given DFA $A = \langle Q, \Sigma, \delta, q_0, F \rangle$, we define the DFA

$$\neg A := \langle Q, \Sigma, \delta, q_0, Q - F \rangle.$$

Prop: For any DFA $A = \langle Q, \Sigma, \delta, q_0, F \rangle$,

$$\overline{L(A)} = L(\neg A).$$

Proof: Let $x \in \Sigma^*$ be any string.

$$x \in \overline{L(A)} \iff x \notin L(A)$$

$\iff$ A rejects x

$\iff$ A ends in a rejecting state $q \notin F$ on input x

$\iff \neg A$ ends in an accepting state (the same q, but now $q \in Q - F$)

$\iff \neg A$ accepts x

$\iff x \in L(\neg A)$

$\therefore$ Since $x \in \Sigma^*$ was an arbitrary string, $\overline{L(A)} = L(\neg A)$ (same members). $\square$

Cor: The complement of a regular language is regular.

Def: REG_Σ is the class of all regular languages over Σ.

Cor says: REG_Σ is closed under complementation.