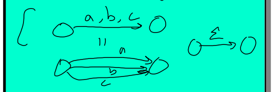
$\Sigma = \{a, b, c\}$

Answer yes on an input string over Σ iff it ends with b

Ex: Automaton that accepts a string over $\{a, b, c\}$ iff it starts with b.

Notation: For alphabet Σ, we let Σ^* denote the set of all strings over Σ.



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

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

where

Q is some finite set (the state set; elements of Q are called states),

Σ is an alphabet (the input alphabet)

$\delta : Q \times \Sigma \longrightarrow Q$ (the transition function)

[in a diagram, $q \xrightarrow{a} r$ means $\delta(q, a) = r$]

$q_0 \in Q$ (the start state)

$\to q_0$

$F \subseteq Q$ (the elements of F are the accepting states; states not in F are the rejecting states)

Tabular form

	a	b	c
$\to$ Ra	Ra	Rb	Rc
*Rb	Ra	Rb	Rc
Rc	Ra	Rb	Rc

transition table for δ

Tabular form $\longleftrightarrow$ transition diagram (digraph)

(you should be able to go either way)

Operational semantics of DFAs ("how they compute")

Def: Fix an alphabet Σ. A language over Σ is any subset of Σ^* (i.e., some set of strings)

Def: Given a DFA A with input alphabet Σ, the language of A, denoted $L(A) = \{x \in \Sigma^* : \underbrace{A \text{ accepts } x}_{\text{need more defs}}\}$.

Def: Given a DFA $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ and a string $x \in \Sigma^*$, the computation path (or trace) of A on input x is the finite sequence of states $s_0, s_1, \ldots, s_n$ $(n = |x|)$ (in Q) such that, writing $x = x_1 x_2 \cdots x_n$ (each $x_i \in \Sigma$),

$s_0 = q_0$

$s_i = \delta(s_{i-1}, x_i)$ (for all i, $1 \le i \le n$)

s_n is the <u>final state</u> of the path.

<u>Note</u>: $\forall$ DFA A & input x, the comp path of A on x is uniquely determined.

<u>Def</u>: For DFA A & string x over A's input alphabet, A <u>accepts</u> x just when the final state of A's comp. path on x is in an accepting state of A. Otherwise, we say that A <u>rejects</u> x.

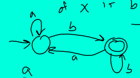
<u>Def</u>: The language of a DFA A, $L(A)$ — we say that A <u>recognizes</u> $L(A)$

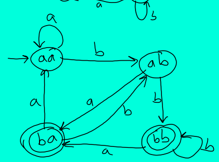
So, A <u>recognizing</u> a language L means A accepts all strings in L and rejects all strings not in L.

<u>Ex</u>: $\Sigma = \{a, b\}$

A DFA A such that

$$L(A) = \{x \in \Sigma^* : |x| \geq 2 \text{ \& the penultimate symbol of } x \text{ is } b\}$$





<u>Def</u>: A language $L \subseteq \Sigma^*$ is <u>regular</u> if $L = L(A)$ for some DFA A.

DFA constructions to show closure properties of the class of reg. langs.

<u>Complement</u>: Given a DFA $A = \langle Q, \Sigma, \delta, q_0, F \rangle$, define

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

so — a state is accepting in $\neg A$ iff it is rejecting in A.

<u>Prop</u>: For any DFA A,

$$L(\neg A) = \underbrace{\Sigma^*}_{A\text{'s input alphabet}} - L(A)$$

Write $\overline{L(A)}$ for $\Sigma^* - L(A)$ when Σ is assumed.

<u>Proof</u>: Given DFA $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ and $x \in \Sigma^*$, let $s \in Q$ be the final state of A's comp path on input x. Since s does not depend on the set F, s is also the final state of $\neg A$'s comp path on x. Then

$\neg A$ accepts x

$\iff s \in Q - F$

$\iff s \notin F$

$\iff A$ rejects x

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

Thus $L(\neg A) = \overline{L(A)}$, because this holds for all x. ▨

<u>Cor</u>: If $L \subseteq \Sigma^*$ is regular, then so is $\overline{L}$.

<u>Prop</u>: If L_1 & $L_2 \subseteq \Sigma^*$ are regular, then so is $L_1 \cap L_2$