<u>Def</u>: Fix alphabet Σ.

$\Sigma^* := \{ x \mid x \text{ is a string over } \Sigma \}$.

A <u>language</u> (over Σ) L is an arbitrary set of strings over Σ: $L \subseteq \Sigma^*$.

<u>Def</u>: A <u>deterministic finite automaton</u> (DFA) is a tuple $\langle Q, \Sigma, \delta, q_0, F \rangle$ where:

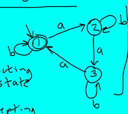
- Q is a finite set (the <u>state set</u>; elts of Q are <u>states</u>).
- Σ is an alphabet (the <u>input alphabet</u> — all inputs to the DFA are strings over Σ)
- δ (later)
- $q_0 \in Q$ (the <u>start state</u>)
- $F \subseteq Q$ (elements of F are <u>accepting</u>; elements of $Q - F$ are <u>rejecting</u>)
- $\delta : Q \times \Sigma \longrightarrow Q$ (the <u>transition function</u>)

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

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

Ex:

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



Trans. diagram

○ — rejecting state

◎ — accepting state

$Q = \{1, 2, 3\}$

$\delta(2, a) = 3$

$\delta(1, b) = 1$

$q_0 = 1$

$F = \{1\}$

Formally:

$\langle \{1,2,3\}, \{a,b\}, \delta, 1, \{1\} \rangle$

where

δ:

states \ symbols	a	b
1	2	1
2	3	2
3	1	3

Tabular form:

	a	b
→ *1	2	1
2	3	2
3	1	3

Informally, given a DFA $A = \langle Q, \Sigma, \delta, q_0, F \rangle$, and an input string $x \in \Sigma^*$, A <u>accepts</u> x if A winds up (ends) in accepting state after reading x.

More formally:

<u>Def</u>: $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ is a DFA and

$x = x_1 x_2 \cdots x_n \quad (n \geq 0)$

$x_i \in \Sigma \quad (\forall i, 1 \leq i \leq n)$

$(x \in \Sigma^*)$

A (<u>computational</u>) <u>trace</u> of A on input x is a sequence $s_0, s_1, \ldots, s_n \in Q$ of states such that:

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

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

Note: every DFA/input combo has a <u>unique</u> trace.

Say a trace $s_0, \ldots, s_n$ <u>ends in</u> state s_n.

<u>Def</u>: Given A as above and $x \in \Sigma^*$. Say that A <u>accepts</u> x if the comp. trace of A on input x ends in a accepting state of A. Otherwise, A <u>rejects</u> x.

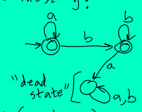
<u>Def</u>: A as above. The <u>language of A</u> (denoted $L(A)$) is

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

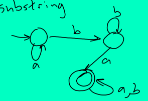
$L(A)$ is the language <u>recognized</u> by A.

Ex[prev]: $L(A)=\{x\in\{a,b\}^* :$ x has a multiple of 3 many a's (& any number of b's)$\}$

Ex: DFA recognizing lang. of strings over $\{a,b\}$ that don't have ba as a substring:



Ex: (complement):
lang. of all string over $\{a,b\}$ that do have ba as a substring



same DFA as before, but swap accepting states with rejecting states.

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

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

Prop: If L is recognized by some DFA, then $\overline{L}$ is recognized by the DFA obtained by swapping accept & reject states.

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

Prop (restated): For any DFA $A = \langle Q,\Sigma,\delta,q_0,F\rangle$,
$$\overline{L(A)} = L(\neg A)$$

Proof: Let $x\in\Sigma^*$ be arbitrary, and let $s_0,\ldots,s_n$ be the computational trace of A on x. Then

$x\in\overline{L(A)}$
$\iff x\notin L(A)$
$\iff A$ rejects x
$\iff s_n\notin F$
$\iff s_n\in Q-F$
$\iff \neg A$ accepts x
[A and ¬A have the same computational trace on x]
$\iff x\in L(\neg A)$.

Since $x\in\Sigma^*$ was arbitrary,
$$\overline{L(A)} = L(\neg A) \quad \blacksquare$$

Def: Let $L\subseteq\Sigma^*$ be any language. We say that L is regular if $L=L(A)$ for some DFA A.

Ex: $\{x\in\{a,b\} :$ x contains ba as a substring$\}$ is regular.

$\{x : x$ has an odd # of b's$\}$ is regular.

⋮

Cor: If L is regular, then $\overline{L}$ is regular.

Def: REG_Σ (Σ alphabet) is the class of all regular language over Σ.

Ex: $L=\{x :$ x has an odd number of a's and ends with b$\}$

$L=L_1\cap L_2$ where
$L_1 := \{x : x$ has odd # of a's$\}$
$L_2 := \{x : x$ ends with b$\}$

$L_1 = L(A_1)$ where

$A_1 =$ 

A_2