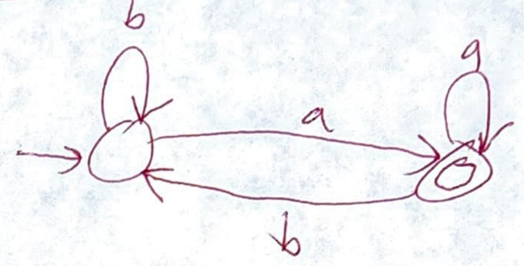
CSCE 355 1/14/2026

<u>Recall</u>: A DFA is a "machine" that has

- a finite set of states
- a start state
- set of accepting states
- transition function

A =



<u>Definition</u>: A <u>deterministic</u> <u>finite</u> <u>automaton</u> (DFA) is a 5-tuple $\langle Q, \Sigma, \delta, q_0, F \rangle$ where

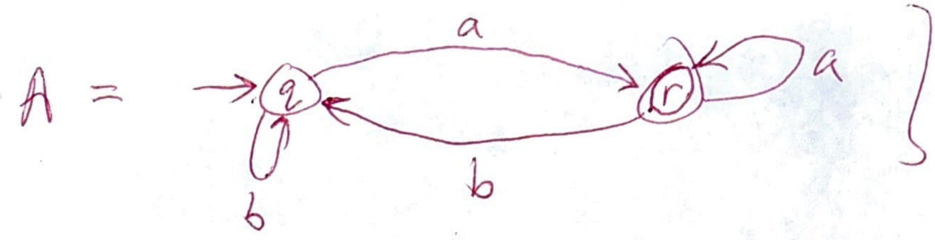
- Q is a finite set (elements of Q are called <u>states</u>)
- Σ is an alphabet (the <u>input alphabet</u>; inputs are strings over Σ)
- $q_0 \in Q$ (the <u>start state</u>)
- $F \subseteq Q$ (elements of F are called <u>accepting</u> states; elements of $Q - F$ are <u>rejecting</u> states)
- $\delta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

($q \in Q$, $a \in \Sigma$, $\delta(q, a) = r$ means

$$q \xrightarrow{a} r$$
)

So in the transition diagram, there is exactly one edge leaving q labeled a ($\forall q \in Q, \forall a \in \Sigma$)

(2)

$A =$  transition diagram

<u>Tabular form</u>: table for the δ-function, with some annotations:

$$A = \langle \{q, r\}, \{a, b\}, \delta, q, \{r\} \rangle$$

$A =$

	a	b
$\to q$	r	q
$* r$	r	q

$\to$ start state

$*$ means accepting state

A DFA gives just a binary result for each input string: accept or reject, depending on the state <u>reached upon</u> reading the entire input string.

<u>Operational semantics</u> of a DFA (i.e., how it computes)

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

$$\hat{\delta} : Q \times \Sigma^* \longrightarrow Q$$

$\forall q$: $\hat{\delta}(q, \varepsilon) = q$ $\qquad$ $\hat{\delta}(q, ab) = \delta(\delta(q, a), b)$

$\forall a \in \Sigma$ $\hat{\delta}(q, a) = \delta(q, a)$ $\qquad$ $\hat{\delta}(q, abc) = \delta(\delta(\delta(q, a), b), c)$

$$q \xrightarrow{a} r \xrightarrow{b} s \xrightarrow{c} t$$

$$\hat{\delta}(q, abc) = t$$

Formally, we define $\hat{\delta}$ by string induction:

For any $q \in Q$,

$$\longrightarrow \hat{\delta}(q, \varepsilon) := q$$

For any $x \in \Sigma^*$ and symbol $a \in \Sigma$

$$\longrightarrow \hat{\delta}(q, xa) = \delta(\hat{\delta}(q, x), a)$$

$$q \overset{x}{\rightsquigarrow} r \xrightarrow{a} s$$

$r = \hat{\delta}(q, x)$

$s = \delta(r, a)$

This defines $\hat{\delta}$ uniquely by string induction.

<u>Facts:</u> $\forall q \in Q, \forall a \in \Sigma, \hat{\delta}(q, a) = \delta(q, a)$

<u>Proof</u> (by string induction)

$$\hat{\delta}(q, a) = \hat{\delta}(q, \varepsilon a)$$

$$= \delta(\underbrace{\hat{\delta}(q, \varepsilon)}_{q}, a)$$

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

$[\ \varepsilon x = x \varepsilon = x\ ]$

<u>Fact:</u> $\forall q \in Q, \forall x, y \in \Sigma^*, \hat{\delta}(q, xy) = \hat{\delta}(\hat{\delta}(q, x), y)$

Proof in by string induction on y //

<u>Def</u>: Let $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ be a DFA and let $x \in \Sigma^*$ be a string over Σ.

Say that <u>A accepts x</u> if $\hat{\delta}(q_0, x) \in F$.

Otherwise, say that <u>A rejects x</u>.

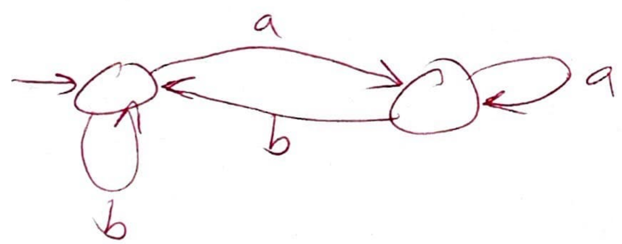
The <u>language recognized by A</u> is the

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

(Recall a language over Σ is any subset of Σ^*, i.e., any set of strings over Σ)

set of all strings (incl. ε) over Σ

A =



$$L(A) = \{x \in \{a,b\}^* : x \text{ ends in } a\}$$

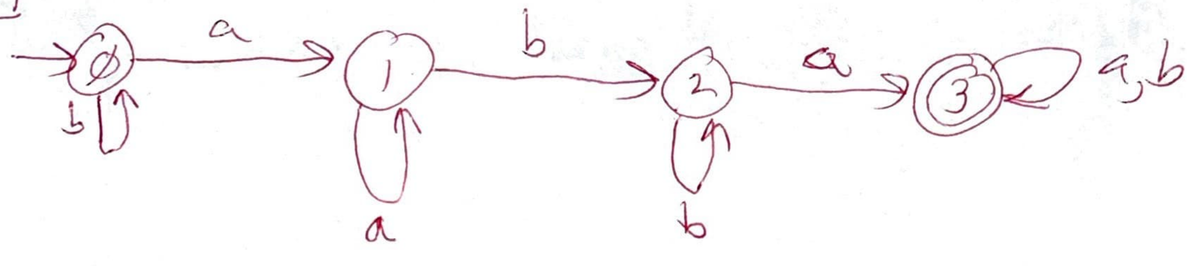
$$= \{x \in \{a,b\}^* : (\exists y \in \{a,b\}^*)[x = ya]\}$$

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

Ex: $L = \{ x \in \{a,b\}^* :$ there is a <u>b</u> somewhere between ~~to~~ two <u>a</u>'s in x, not necessarily contiguously $\}$

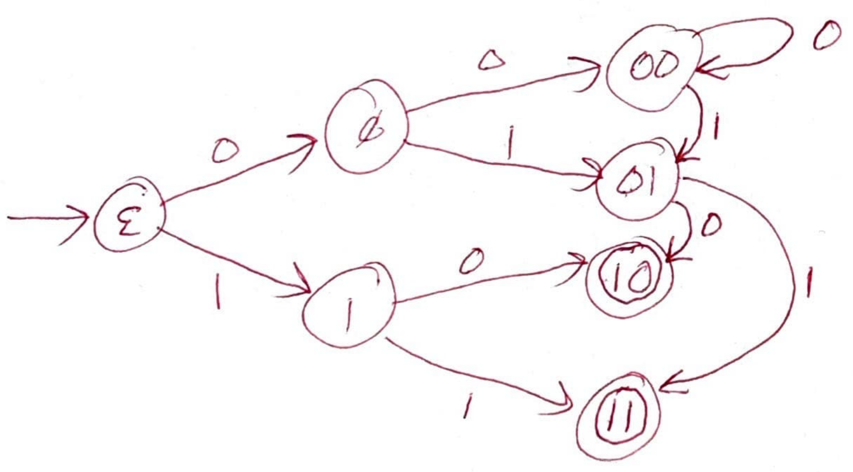
Tabular form

	a	b
$\rightarrow \emptyset$	1	$\emptyset$
1	1	2
2	3	2
*3	3	3



Ex: ~~A~~ $\Sigma = \{0,1\}$

A DFA such that $L(A) = \{ x \in \{0,1\}^* :$ x's 2nd to last symbol is $1 \}$.



Finish the transition diagram!

<u>Bonus question</u>: Can you do this with fewer than 7 states? If so, how? Spoiler: yes — 4 states

DFA constructions: combining/modifying DFAs to make new DFAs.

Complement:

Given $A = \langle Q, \Sigma, \delta, q_0, F \rangle$,

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

Then if A recognizes L $(L \subseteq \Sigma^*)$,

then $\neg A$ recognizes $\overline{L}$ $(= \Sigma^* - L)$

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

Proof: Let $L = L(A)$ for DFA $A = \langle Q, \Sigma, \delta, q_0, F \rangle$

Then for any $x \in \Sigma^*$,

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

$\iff \hat{\delta}(q_0, x) \in F$

$\iff \hat{\delta}(q_0, x) \notin Q - F$

$\iff \neg A$ rejects x (A & $\neg A$ share the same start state & $\hat{\delta}$ function)

$\iff$ ~~$x \notin L(A$~~ $x \notin L(\neg A)$

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