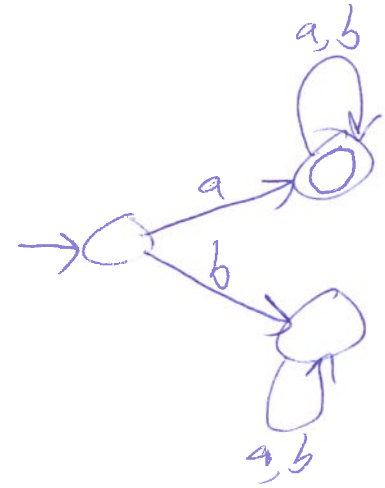
CSCE 355
1/15/2025

Automata



Recall: An automaton has
- a set of states
- an alphabet ex: ($\Sigma = \{a, b\}$)
- arrows labeled with alphabet symbols that go between states
- some states are accepting (rest are rejecting)
- designated start state

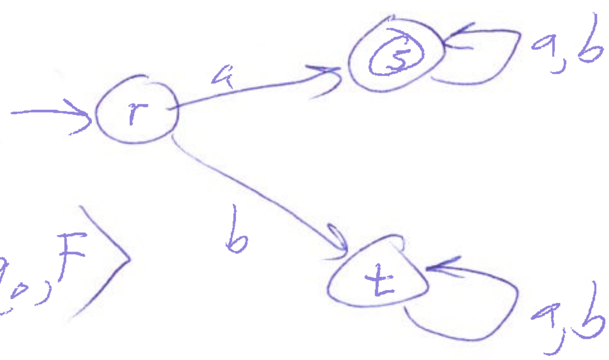
Definition: A deterministic finite automaton (DFA) is a 5-tuple $\langle Q, \Sigma, \delta, q_0, F \rangle$ where
- Q is a finite set (the state set; elements of Q are states)
- Σ is an alphabet (the input alphabet; inputs to the DFA are strings over Σ)
- $\delta : Q \times \Sigma \to Q$ (the transition function)

$q \xrightarrow{a} r$ means $\delta(q, a) = r$

- $q_0 \in Q$ (the start state)
- $F \subseteq Q$ (a subset of Q; elements of F are the accepting states, states in $Q-F$ are the rejecting states)

Example:

$A =$



transition diagram

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

$Q = \{r, s, t\}$

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

$q_0 = r$

$F = \{s\}$

$(Q-F = \{r, t\})$

δ:

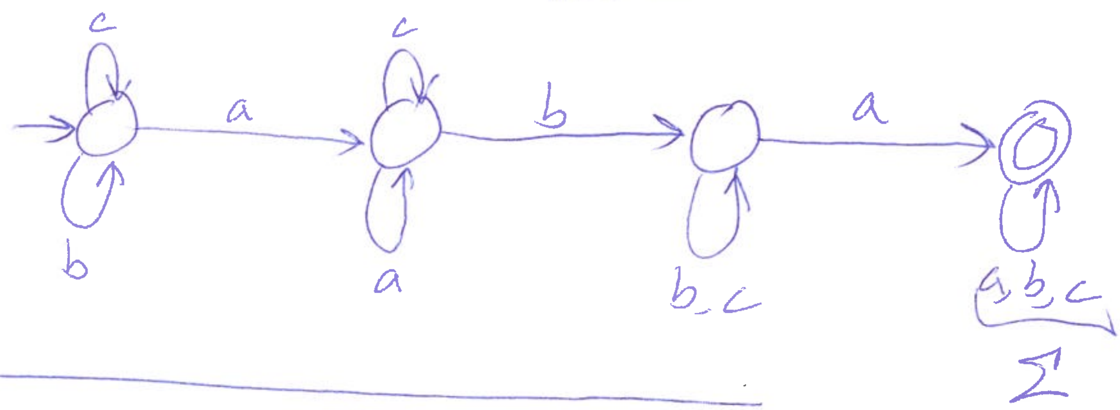
	a	b
$\rightarrow r$	s	t
$*s$	s	s
t	t	t

Tabular form

You should be able to go back & forth between transition diagram & tabular form

Ex: $\Sigma = \{a, b\}$ want a DFA that accepts a string over Σ just when the string has a b between two a's (not nec. adjacent)

$x = \underbrace{\cdots}\, a \underbrace{\cdots}\, b \underbrace{\cdots}\, a \underbrace{\cdots}$ accept!



Notation: For alphabet Σ, let Σ^* denote the set of all strings (incl ε) over Σ.

Operational Semantics of a DFA

Given a DFA $A = \langle Q, \Sigma, \delta, q_0, F \rangle$, we define the extended transition function

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

Ex: $w = w_1 w_2 \cdots w_n$ (each $w_i \in \Sigma$)

mean

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

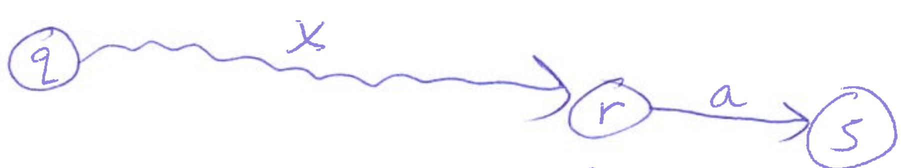
Define $\hat{\delta}$ by string induction:

For any state $q \in Q$:

- $\boxed{\hat{\delta}(q, \varepsilon) = q}$ (no transitions)
- For any $w \in \Sigma^*$, $w \neq \varepsilon$, write $w = xa$ where $x \in \Sigma^*$ is the principal prefix of w and $a \in \Sigma$ is the last symbol.

Assume (inductive hypothesis) that $\hat{\delta}(q, x)$ is well-defined. Then define

$$\hat{\delta}(q, w) := \delta(\hat{\delta}(q, x), a)$$



$r = \hat{\delta}(q, x)$ $\quad$ $s = \delta(r, a) = \hat{\delta}(q, xa) = \hat{\delta}(q, w)$

$\hat{\delta}$ agrees with δ on strings of length 1 (symbols):

$a \in \Sigma$, $q \in Q$

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

<u>Def</u>: Let A be a DFA as above

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

and let $x \in \Sigma^*$ be any string. We say that <u>A accepts x</u> if $\hat{\delta}(q_0, x) \in F$

[The final state of A on input x is accepting].

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

~~The language~~

<u>Def</u>. Fix an alphabet Σ. A <u>language</u> over Σ is any subset of Σ^*

<u>Def</u>: DFA A as above. The <u>language of A</u> (the language <u>recognized</u> by A) is

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

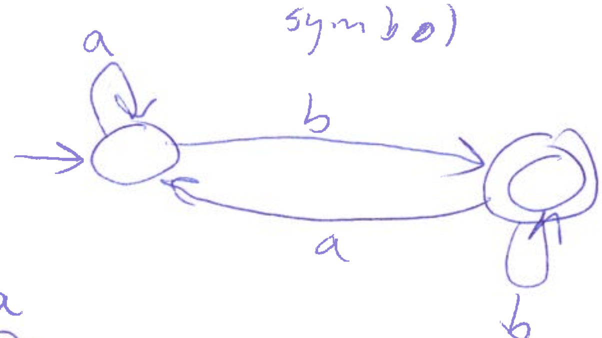
recognizes Σ^*

Def: Let L be a language (over some fixed Σ) we say that L is <u>regular</u> if $L = L(A)$ for some DFA A. (I.e., some DFA recognizes L).

Ex: Want a DFA recognizing this lang:

$$L := \{x \in \{a,b\}^* : |x| \geq 2 \text{ \& its 2nd to last character (symbol) is a } b\}$$

Recall:



accepts iff last symbol of the input is b.