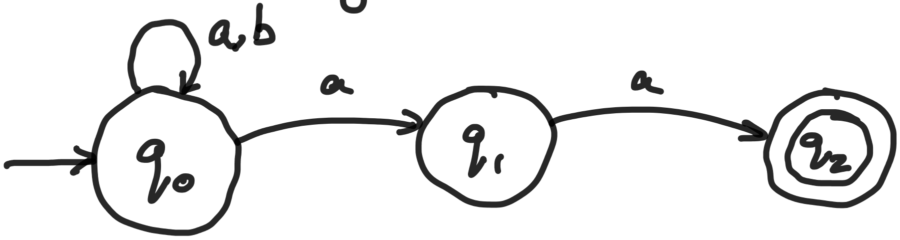
Lecture 3 Nondeterministic Finite Automata

Thursday, January 14, 2021 10:38 AM

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

Example: accept all words ending in "aa".



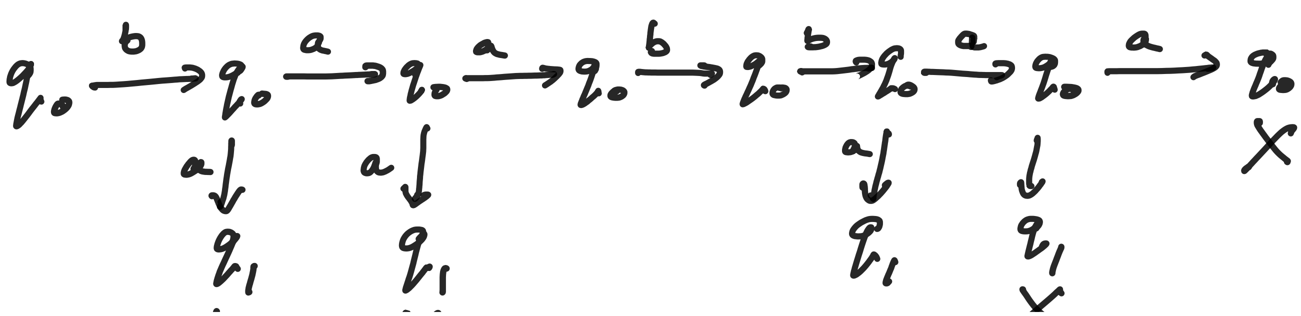
2 possibilities for 'a' in q_0: q_0 or q_1

New mechanism: non deterministic "guessing".

If no transition is indicated the m/c jams and rejects.

There must exist one possible path to acceptance.

e.g.: baabbaa



$a \downarrow q_2$ X

No place to go

$a \downarrow q_2$ ✓

one path is all it takes.

Non-determinism is a <u>design</u> <u>tool</u>

We can always design a DFA that accept exactly the same language as an NFA.

ANOTHER EXTENSION: NFA + ε moves

This machine can jump to another state without reading an input.

$$DFA \equiv NFA \equiv NFA + \varepsilon$$

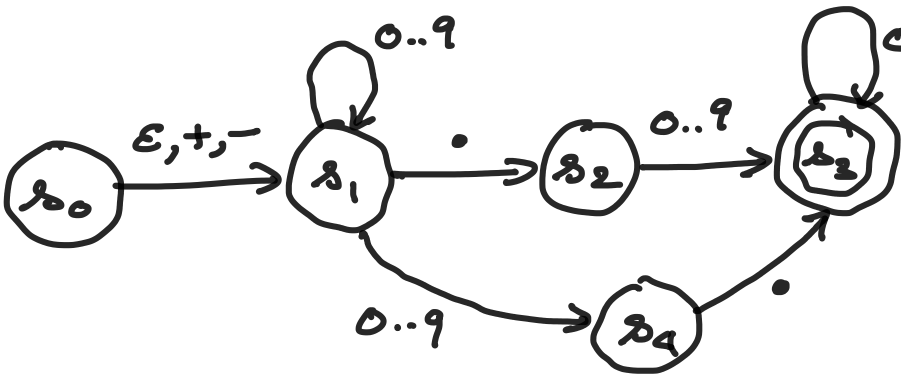
<u>EXAMPLE</u> Recognize decimal numbers

(i) an optional +, − sign (ii) a sequence of digits (iii) a <u>decimal point</u> (iv) a seq of digits

$$\Sigma = \{+, -, \bullet, 0, 1, 2, \cdots 9\}$$

307.3, .406, +97., 0.0, −3.1,

The decimal point is mandatory



Test for yourself : 3.5.5, 12345.0

FORMALIZATION OF $L(M)$

DFA $M = (S, s_0, \delta : S \times \Sigma \to S, F \subseteq S)$.

$\delta(s,a) = s'$

$\delta^* : S \times \Sigma^* \longrightarrow S$

$\forall s;\ \delta^*(s, \varepsilon) = s$

$\delta^*(s, wa) = \delta(\delta^*(s,w), a)$

$L(M)$: language accepted by M

$L(M) = \{ w \in \Sigma^* \mid \delta^*(s_0, w) \in F \}$

N is an NFA; what is $L(N)$?

Formal def of NFA $= (Q, Q_0, \Delta, F \subseteq Q)$

(1) Q : a finite set of states

(2) $Q_0 \subseteq Q$, a <u>set</u> of start states

(4) $F \subseteq Q$, a <u>set</u> of accept states

(3) $\Delta : Q \times \Sigma \longrightarrow 2^Q$ ($\mathcal{P}(Q)$: power set of Q).

(state $\times$ letter) $\longmapsto$ <u>set</u> of possible next states

$\phi \in 2^Q$ so it could be that there is no next state : jam & reject.

$\Delta^* : Q \times \Sigma^* \longrightarrow 2^Q$

$\Delta^*(q, \varepsilon) = \{q\}$

$\left[a \in \Sigma \quad \Delta^*(q, a) = \Delta(q, a) \right]$

$$\Delta^*(q, wa) = \bigcup_{q' \in \underbrace{\Delta^*(q, w)}} \Delta(q', a)$$

$$L(N) = \{ w \in \Sigma^* \mid \exists q_0 \in Q_0, \Delta^*(q_0, w) \cap F \neq \phi \}$$

at <u>least</u> one state where the machine may end up is an accept state.

Thm For any NFA N, there exists a DFA, M s.t $L(M) = L(N)$

Proof: Given $N = (Q, Q_0, \Delta, F)$

I define $M = (S, s_0, \delta, \hat{F})$ where

$S = 2^Q \quad s_0 = Q_0 \in 2^Q = S$

$\hat{F} = \{A \subseteq Q \mid A \cap F \neq \phi\}$

$A \subseteq Q, A \in 2^Q = S, \quad \delta(A, a) = \bigcup_{q \in A} \Delta(q, a)$

This is a DFA, now I have to show

$L(M) = L(N)$

Lemma $\Delta^*(A, \omega) = \delta^*(A, \omega) \quad \forall \omega \in \Sigma^*$

Proof By induction on the length of $|\omega|$

Base $|\omega| = 0$ i.e $\omega = \varepsilon$

$\Delta^*(A, \varepsilon) = A = \delta^*(A, \varepsilon)$

Ind case let $\omega = xa$ &

assume $\forall A \subseteq Q \quad \Delta^*(A, x) = \delta^*(A, x)$

$\delta^*(A, xa) = \delta(\delta^*(A, x), a)$ [Def of δ^*]

$= \delta(\Delta^*(A, x), a)$ [IH]

$= \delta^*(\Delta^*(A, x), a)$ [Def of δ]

$= \Delta(\Delta(A, x), a) \quad [\ldots]$

$= \Delta^*(A, xa)$ ✓ END of LEMMA

Proof of theorem completed:

$L(N) = \{\omega / \Delta^*(Q_0, \omega) \cap F \neq \phi\}$

$= \{\omega / \Delta^*(Q_0, \omega) \in \hat{F}\}$

$= \{\omega / \delta^*(Q_0, \omega) \in \hat{F}\}$ [LEMMA]

$= \{\omega / \delta^*(s_0, \omega) \in \hat{F}\}$

$= L(M)$.

BOTTOM LINE:

DFA, NFA, NFA+ε all have the same expressive power.

If I ask you to prove that something is regular feel free to construct an NFA+ε or an NFA.

I will often ask: Given a regular language L show that some modified language constructed from L is also regular.

e.g. $L = \ldots$

$L/a := \{ w \mid aw \in L \}$

You solve such questions by showing how to construct an NFA (or NFA+ε) for the modified language given a DFA for the original language.