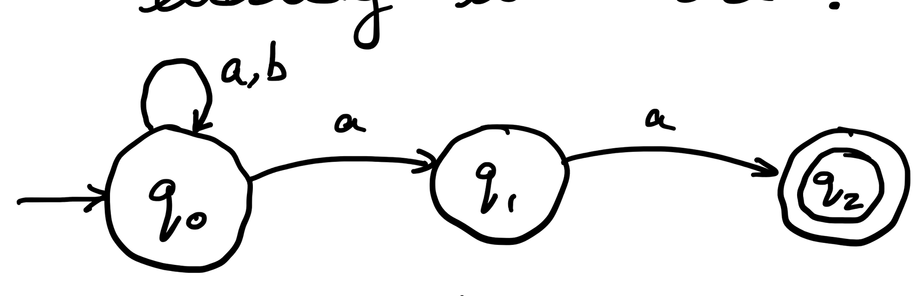
Lecture 3 Nondeterministic Finite Automata

Thursday, January 14, 2021 10:38 AM

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

Example : accept all words ending in "aa".



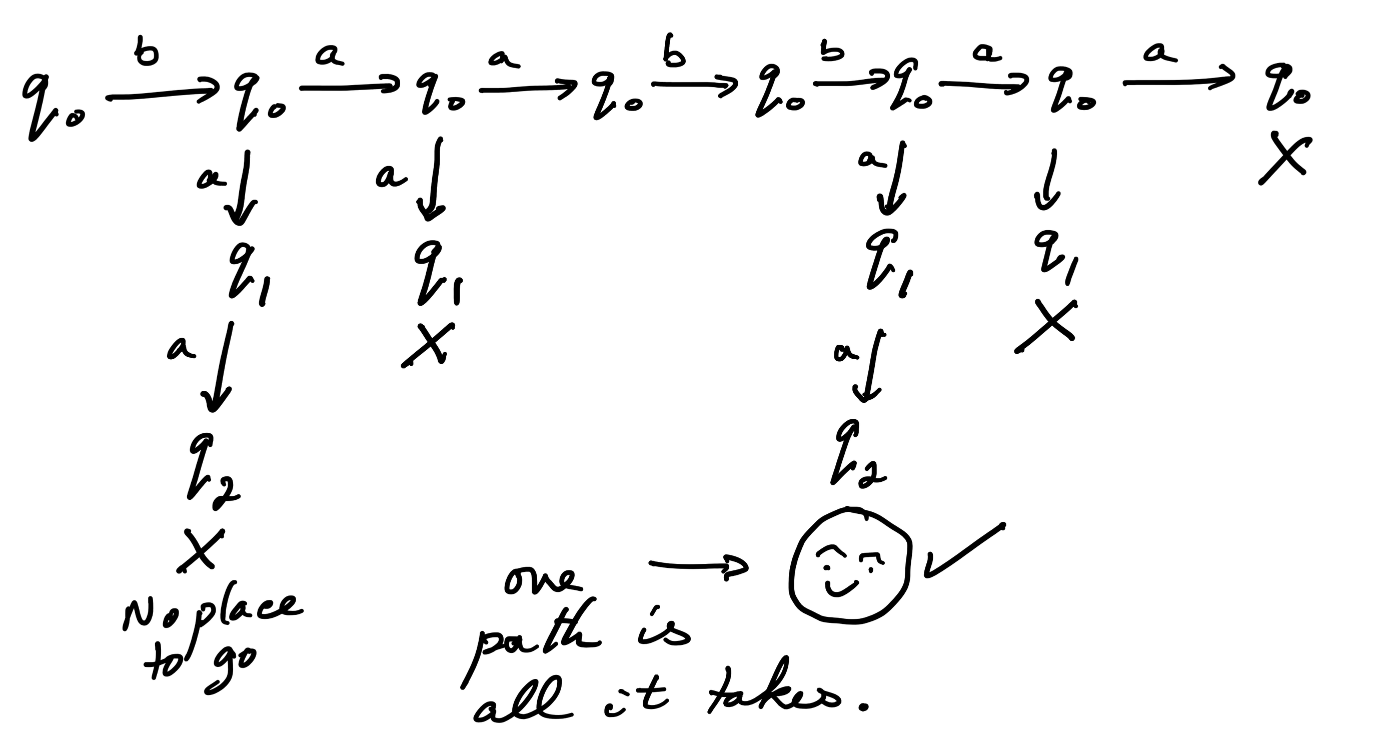
2 possibilities for 'a' in q_0: q_0 or q_1

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



Non-determinism is a design tool

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.

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

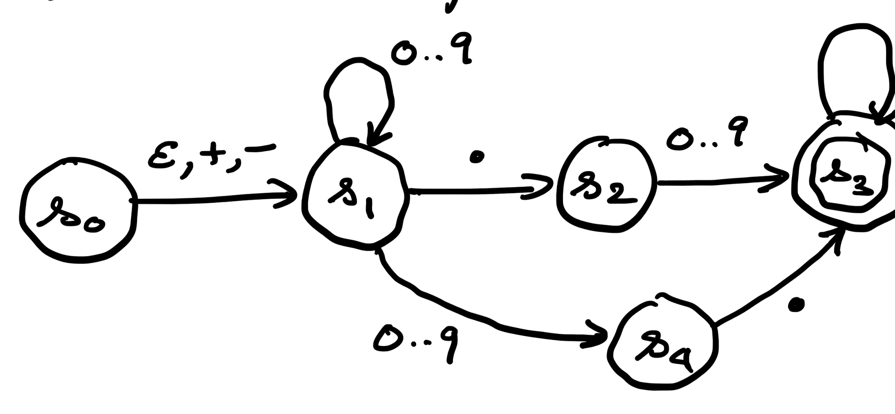
EXAMPLE Recognize decimal numbers

(i) an optional +, − sign (ii) a sequence of digits (iii) a decimal point (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 $M = (S, s_0, \delta : S \times \Sigma \to S, F \subseteq S)$.

$\delta(s, a) = s'$ $\quad s \xrightarrow{a} s'$

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

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

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

$L(M)$: 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 set of start states

(4) $F \subseteq Q$, a set of accept states

(3) $\Delta : Q \times \Sigma \longrightarrow 2^Q$ ($\wp(Q)$: power set of Q).

(state × letter) $\longmapsto$ set of possible next states

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

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

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

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

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

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

at least 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

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

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

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

This is a DFA ; now I have to show

$$L(M) = L(N)$$

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

Proof By induction on the length of $|w|$

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

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

Ind case let $w = xa$ & assume $\forall A \subseteq Q \quad \Delta^*(A, x) = \delta^*(A, x)$

$$\begin{aligned}
\delta^*(A, xa) &= \delta(\delta^*(A, x), a) && [\text{Def of } \delta^*] \\
&= \delta(\Delta^*(A, x), a) && [\text{IH}] \\
&= \Delta^*(\Delta^*(A, x), a) && [\text{Def of } \delta] \\
&= \Delta^*(A, xa) && \checkmark\ \text{END of LEMMA}
\end{aligned}$$

Proof of thm completed :

$$\begin{aligned}
L(N) &= \{w \mid \Delta^*(Q_0, w) \cap F \neq \phi\} \\
&= \{w \mid \Delta^*(Q_0, w) \in \hat{F}\} \\
&= \{w \mid \delta^*(Q_0, w) \in \hat{F}\} && [\text{LEMMA}] \\
&= \{w \mid \delta^*(s_0, a) \in \hat{F}\} \\
&= L(M).
\end{aligned}$$

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. $L = \cdots$

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

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.