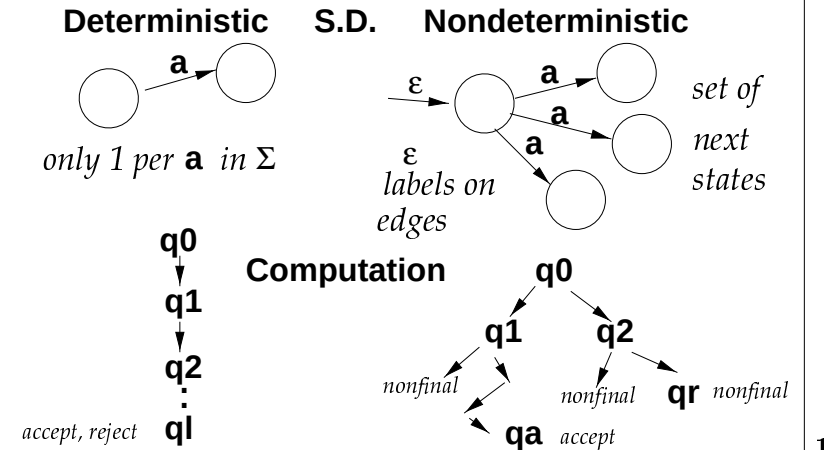


Nondeterministic Finite Automata

Nondeterminism allows several possible next states (no one state is determined)



1

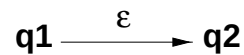
NFA vs. DFA

NFA are seemingly more powerful (NOT!)

NFA allow choice of next state

When choice, NFA may be more compact

Handling ϵ



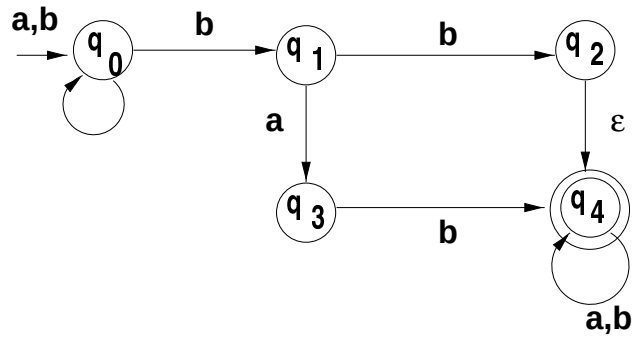
If in q_1 reading input symbol a , and take ϵ edge, still reading a , but in state q_2 .

All ϵ edges must be included in execution tree.

Qu: Is every DFA an NFA?

2

Example NFA

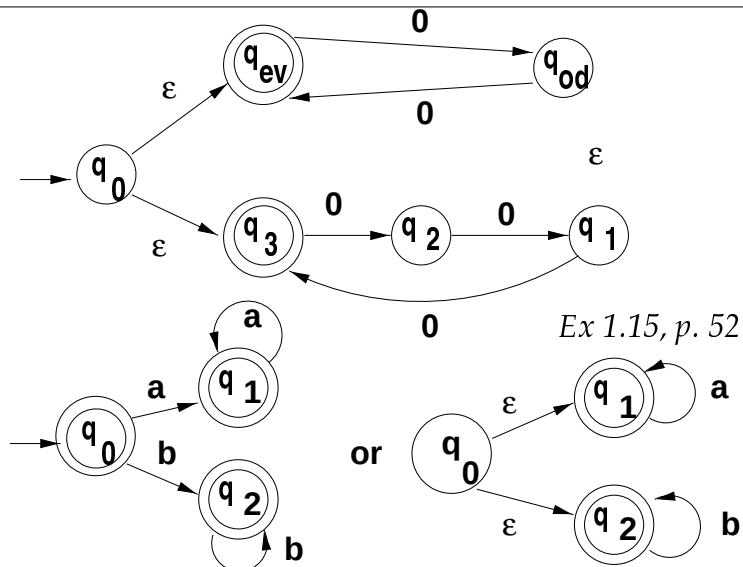


Execution on baabb

$L(M) =$

3

More Example NFA's



4

Formal Definition of Nondeterministic FA

A NFA M is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ where,

1. Q is a finite set of states
2. Σ is a finite set called the *alphabet*
3. $\delta : Q \times \Sigma_{\epsilon} \rightarrow P(Q)$ (power set of Q , include ϵ)
4. q_0 in Q , the *start state*
5. $F \subseteq Q$, set of *accept (final) states*

We say that M *accepts* w

if there is a sequence of states $r_0, r_1, \dots, r_m$ in Q

and $w = y_1 y_2 \dots y_m$, y_i in Σ_{ϵ} such that

1. $r_0 = q_0$
2. r_{i+1} element of $\delta(r_i, y_{i+1})$
3. r_m is in F

5

Equivalence of NFA and DFA

Def: Two NFA M_1, M_2 are *equivalent* if $L(M_1) = L(M_2)$.

Th.: For each NFA, there is an equivalent DFA.

Proof idea:

Start with NFA N . Want DFA M with $L(N) = L(M)$.

We will simulate NFA with DFA.

NFA has *set* of possible next states; DFA just has 1.

Solution: *state* of the DFA will be subset of states of NFA

Proof: Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA.

Case 1: N has no ϵ transitions.

$M = (Q', \Sigma, \delta', q'_0, F')$

1. $Q' = P(Q)$
2. $\delta'(R, a) = \bigcup_{r \in R} \delta(r, a)$
3. $q'_0 = \{q_0\}$
4. $F' = \{R \text{ in } Q' \mid R \cap F \neq \emptyset\}$

6

Equivalence of NFA and DFA

Case 2: N has ϵ transitions.

Proof Idea: What needs to change? Have to account for all ϵ moves after read an input symbol.

Proof: For R a subset of Q , define

$E(R) = \{q \mid q \text{ can be reached from some state in } R \text{ by following only } \epsilon \text{ edges}\}$

Change the construction in Case 1 as follows:

2. $\delta'(R, a) = \{q \text{ in } Q \mid q \text{ in } E(\delta(r, a)) \text{ for some } r \text{ in } R\}$
3. $q'_0 = \{E(q_0)\}$

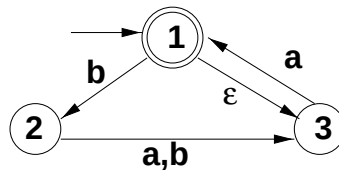
M keeps track of exactly the subset of states N would be in, and accepts when N accepts.

7

Equivalence of NFA and DFA

Cor.: A language is regular iff some NFA recognizes it.

Example: NFA, construct DFA (Ex. 1.21 p. 57)



Get lots of states....can we do better?

States that can't be reached from start

states that have no incoming edges.....More later!

8

Back to: Regular languages are closed under ...

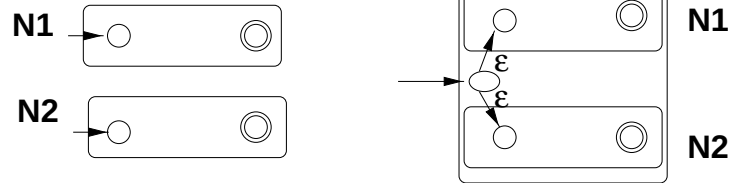
Th. The class of reg. lang. are closed under $\cup$

Let $N1 = (Q1, \Sigma, \delta_1, q1, F1)$,

let $N2 = (Q2, \Sigma, \delta_2, q2, F2)$ be NFA's.

Construct NFA N such that $L(N) = L(N1) \cup L(N2)$

Proof Idea:



Proof: We define NFA N as follows....

9

Regular languages are closed under Union

$N = (Q, \Sigma, \delta, q0, F)$

1. $Q = \{q0\} \cup Q1 \cup Q2$ ($q0$ new, $Q1$ and $Q2$ disjoint)

2. $q0$ is start

3. $F = F1 \cup F2$

4.

$$\delta(q, a) = \begin{cases} \delta_1(q, a) & \text{if } q \text{ in } Q1 \\ \delta_2(q, a) & \text{if } q \text{ in } Q2 \\ \{q1, q2\} & \text{if } q = q0 \text{ and } a = \epsilon \\ \phi & \text{if } q = q0 \text{ and } a \neq \epsilon \end{cases}$$

10

Regular languages are closed under Star

Proof Idea:



Proof: Define $N = (Q, \Sigma, \delta, q_0, F)$ from N_1 .

1. $Q = \{q_0\} \cup Q_1$ q_0 is new

2. q_0 is start

3. $F = \{q_0\} \cup F_1$

4.

$$\delta(q, a) = \begin{cases} \delta_1(q, a) & \text{if } q \text{ in } Q_1, q \text{ not in } F_1 \\ \delta_1(q, a) & \text{if } q \text{ in } F_1, a \text{ not } = \epsilon \\ \delta_1(q, a) \cup \{q_1\} & \text{if } q \text{ in } F_1 \text{ and } a = \epsilon \\ \{q_1\} & \text{if } q = q_0, a = \epsilon \\ \phi & \text{if } q = q_0, a \text{ not } = \epsilon \end{cases}$$

11

Regular languages are closed under Concatenation

Proof Idea: Start out in N_1 , and in any final state of N_1 , can switch to N_2 ; accept if wind up in final state in N_2 .

Proof: Define $N = (Q, \Sigma, \delta, q_1, F)$ from N_1 and N_2

1. $Q = Q_1 \cup Q_2$

2. q_1 is start (same as start of N_1)

3. $F = F_2$

4.

$$\delta(q, a) = \begin{cases} \delta_1(q, a) & \text{if } q \text{ in } Q_1, q \text{ not in } F_1 \\ \delta_1(q, a) & \text{if } q \text{ in } F_1, a \text{ not } = \epsilon \\ \delta_1(q, a) \cup \{q_2\} & \text{if } q \text{ in } F_1 \text{ and } a = \epsilon \\ \delta_2(q, a) & \text{if } q \text{ in } Q_2 \end{cases}$$

12