Nondeterminism and Epsilon Transitions

Mridul Aanjaneya

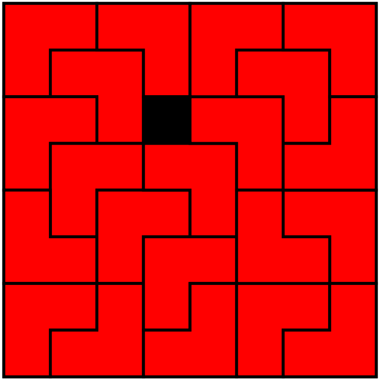
Stanford University

June 28, 2012

Challenge Problem

Question

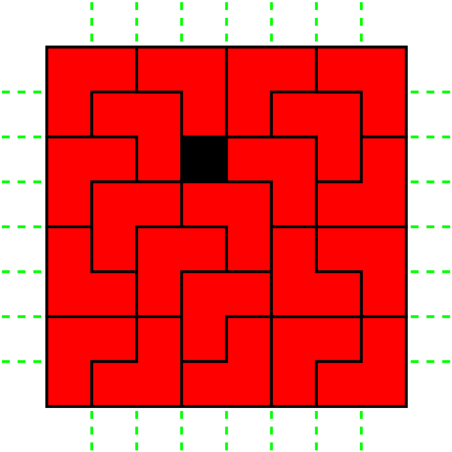
Prove that any square with side length a power of 2, and one square removed, is tileable with L's.



Something to think about ...

Question

Are such tilings always possible?



Recap: Deterministic Finite Automata

Definition

A DFA is a 5-tuple $(Q, \Sigma, \delta_D, q_0, F)$ consisting of:

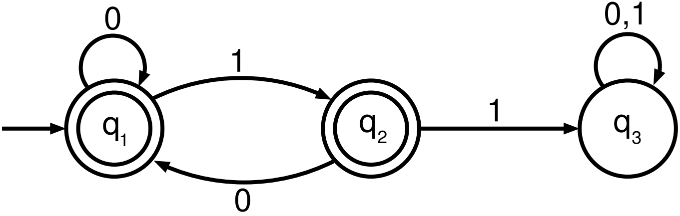
- A finite set of states Q,
- A set of input alphabets Σ,
- A transition function $\delta_D : Q \times \Sigma \to Q$,
- A start state q_0, and
- A set of accept states $F \subseteq Q$.

The transition function δ_D:

- Takes two arguments, a state q and an alphabet a.
- δ_D(q,a) = the state the DFA goes to when it is in state q and the alphabet a is received.

Recap: Graph representation of DFA's

- Nodes correspond to states.
- Arcs represent transition function.
  - Arc from state p to state q labeled by all those input symbols that have transitions from p to q.
- Incoming arrow from outside denotes start state.
- Accept states indicated by double circles.

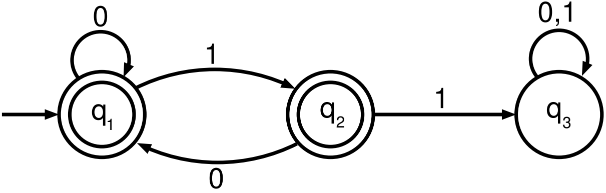


- Accepts all binary strings without two consecutive 1's.

Recap: Transition table for DFA's

		0	1
$\rightarrow$	q_1^*	q_1	q_2
	q_2^*	q_1	q_3
	q_3	q_3	q_3

- A row for each state, a column for each alphabet.
- Accept states are starred.
- Arrow for the start state.



Recap: Regular languages

Definition

A DFA $M = (Q, \Sigma, \delta_D, q_0, F)$ accepts w if there exists a sequence of states $r_0, r_1, \ldots, r_n$ in Q with three conditions:

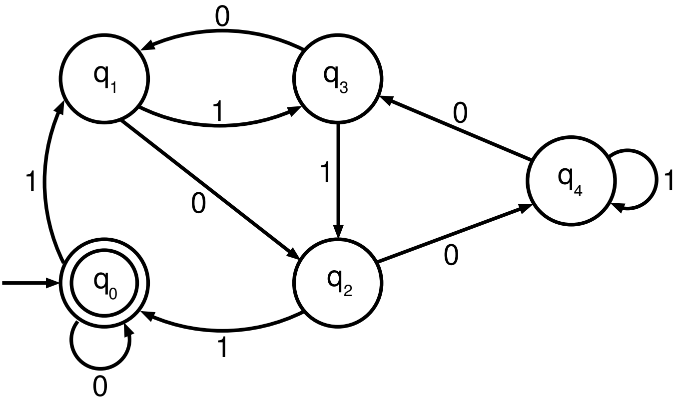
- $r_0 = q_0$
- $\delta_D(r_i, w_{i+1}) = r_{i+1}$ for $i = 0, \ldots, n-1$, and
- $r_n \in F$

- Condition 1 says that M starts in the start state $\mathsf{q_0}$.
- Condition 2 says that M follows δ_D between two states.
- Condition 3 says that last state is an accept state.
- We say that M recognizes L if $\mathsf{L} = \{\mathsf{w} \mid \mathsf{M} \text{ accepts } \mathsf{w}\}$.

Regular Languages: Examples

Example

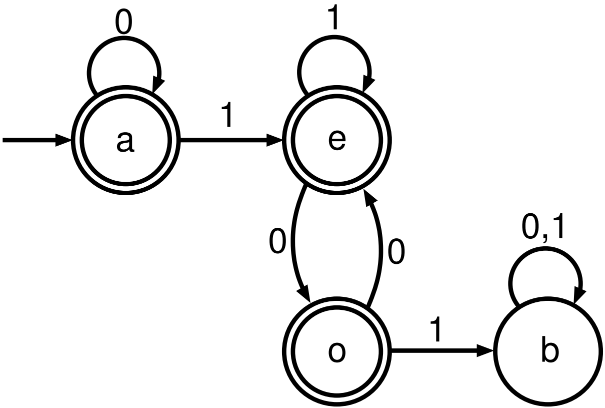
Let $L = \{w \mid w \in \{0,1\}^*$ and w, viewed as a binary integer, is divisible by $5.\}$



Regular Languages: Examples

Example

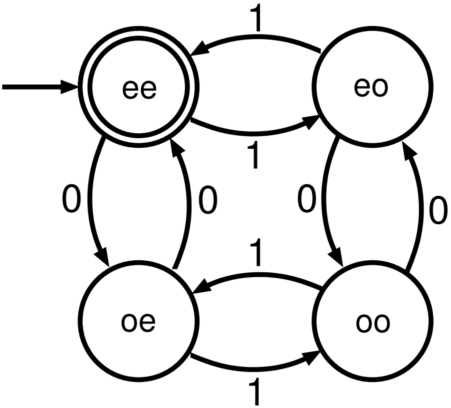
Show that the language of all strings over $\{0, 1\}$ that do not contain a pair of 1's that are separated by an odd number of 0's is regular.



Regular Languages: Examples

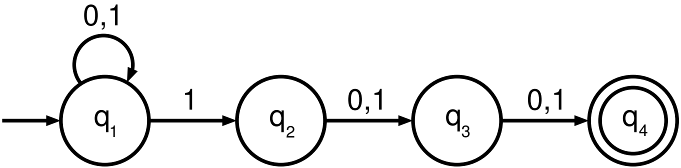
Example

Show that the language of all strings over $\{0, 1\}$ that contain an even number of 0's and 1's is regular.



Nondeterminism

- Deterministic finite automata can only be in one state at any point in time.
  - Recall the definition of the transition function δ_D.
- In contrast, nondeterministic finite automata (NFA's) can be in several states at once!
  - The transition function δ_N is a one-to-many function.

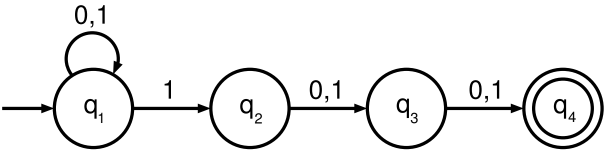


- This NFA recognizes strings in $\{0,1\}^*$ containing a 1 in the third position from the end.

Transition table

		0	1
$\rightarrow$	q_1	q_1	q_1, q_2
	q_2	q_3	q_3
	q_3	q_4	q_4
	q_4^*	q_4	q_4

- q_1 is a nondeterministic state with a one-many transition on 1.

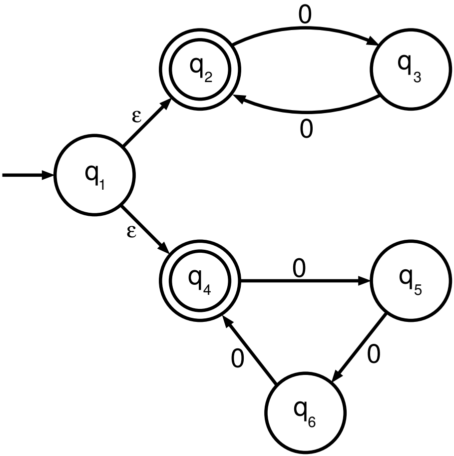


- Intuitively, the NFA always guesses right.

Nondeterministic Finite Automata

Example

An NFA that accepts all strings of the form 0^k where k is a multiple of 2 or 3.



Nondeterministic Finite Automata

Definition

An NFA is a 5-tuple $(Q, \Sigma, \delta_N, q_0, F)$ consisting of:

- A finite set of states Q,
- A set of input alphabets Σ,
- A transition function $\delta_N : Q \times \Sigma_\varepsilon \to \mathcal{P}(Q)$,
- A start state q_0, and
- A set of accept states $F \subseteq Q$.

- Here, Σ_ε denotes the set $\Sigma \cup \{\varepsilon\}$.
- $\mathcal{P}(Q)$ denotes the power set of Q.

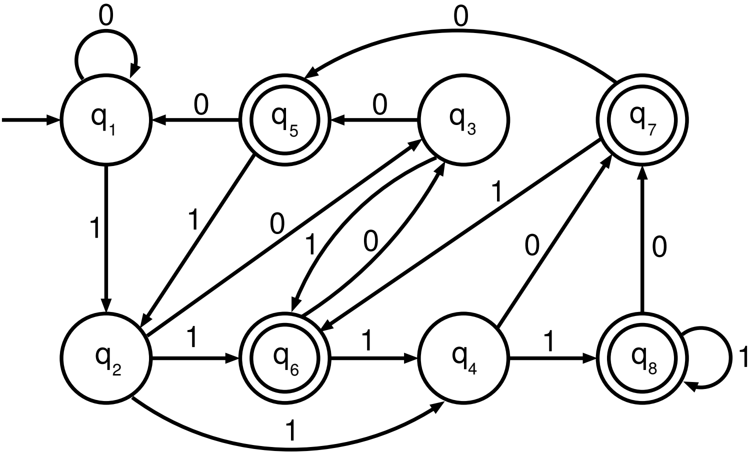
Transition function of an NFA

- δ_N(q,a) is a set of states.
- Extend to strings as follows:
  - **Basis:** δ_N(q,ε) = q
  - **Induction:** δ_N(q,wa) = the union over all states of δ_N(p,a), where p $\in$ δ_N(q,w).
- A string is accepted by an NFA if δ_N(q_0,w) contains at least one state p $\subseteq$ F.
- The language of an NFA is the set of strings it accepts.

Equivalence of DFA's and NFA's

- Every DFA is also an NFA by definition.
  - There is simply no nondeterminism.
- Surprisingly, for every NFA, there is also an equivalent DFA!
  - Two equivalent machines recognize the same language.
  - Nonintuitive, as we'd expect NFA's to be more powerful.
  - Useful, as describing an NFA is much simpler.
- Proof is the subset construction.
- The number of states of the DFA can be exponential in the number of states of the NFA.
- Thus, NFA's accept exactly the regular languages.

- DFA for recognizing strings with a 1 in the third last position.



Subset construction

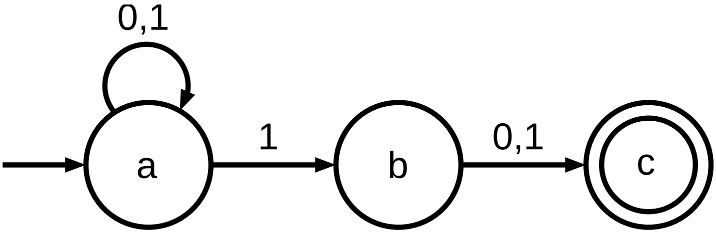
- Given an NFA (Q,Σ,δ_N,q_0,F), construct equivalent DFA with:
  - States $\mathcal{P}(Q)$ (set of subsets of Q).
  - Inputs Σ.
  - Start state $\{q_0\}$.
  - Final states $=$ all those with a member of F.

Note:

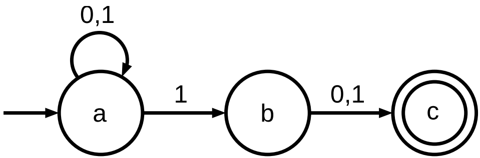
- The DFA states have names that are sets of NFA states.
- But as a DFA state, an expression like $\{p,q\}$ must be read as a single symbol.

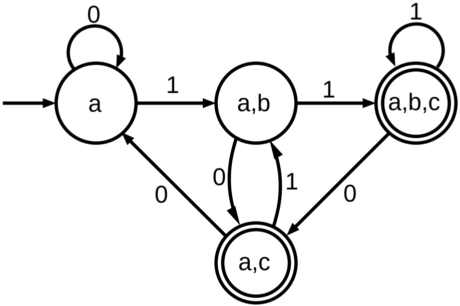
Subset construction

- **Example:** We'll construct the DFA for the following NFA which recognizes strings in $\{0,1\}^*$ containing a 1 in the second position from the end.



Subset construction



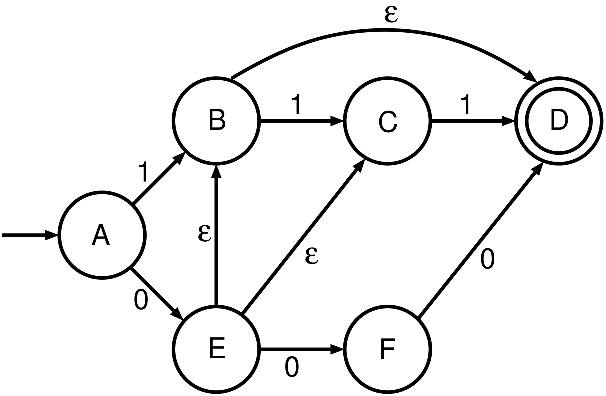


Proof of equivalence: subset construction

- Show by induction on length of w that $\delta_N(q_0, w) = \delta_D(\{q_0\}, w)$
- **Basis:** $w = \varepsilon$. $\delta_N(q_0, \varepsilon) = \delta_D(\{q_0\}, \varepsilon) = \{q_0\}$.
- **Inductive step:** Assume **IH** is true for all strings shorter than w. Let $w = xa$, then **IH** is true for x.
    - Let $\delta_N(q_0, x) = \delta_D(\{q_0\}, x) = S$.
    - Let T = the union over all states p in S of $\delta_N(p, a)$.
    - Then $\delta_N(q_0, w) = \delta_D(\{q_0\}, w) = T$ (by definition).

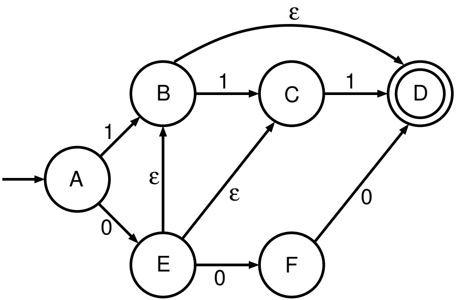
NFA's with ε-transitions

- State-to-state transitions on ε input.
- These transitions are spontaneous, and do not consider the input string.



Closure of States

- CL(q) = set of states that can be reached from state q following only arcs labeled ε.

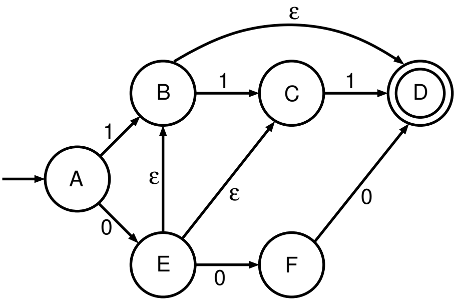


- CL(A) = {A}, CL(E) = {B, C, D, E}.
- Closure of set of states = union of closure of each state.

Extended transition function

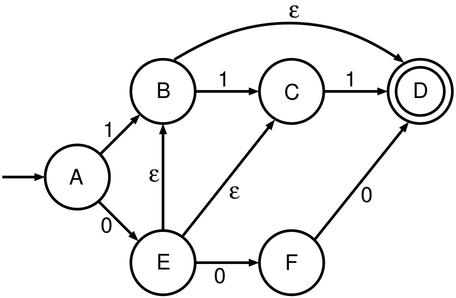
- **Basis:** $\delta_E(q,\varepsilon) = CL(q)$.
- **Induction:** $\delta_E(q,xa)$ is computed as follows:
  1. Start with $\delta_E(q,x) = S$.
  2. Take the union of $CL(\delta(p,a))$ for all p in S.
- **Intuition:** $\delta_E(q,w)$ is the set of states you can reach from q following a path labeled w with ε's in between.

Example: Extended transition function



- $\delta_E(A, \varepsilon) = \mathrm{CL}(A) = \{A\}$.
- $\delta_E(A, 0) = \mathrm{CL}(E) = \{B, C, D, E\}$.
- $\delta_E(A, 01) = \mathrm{CL}(C, D) = \{C, D\}$.

Example: Extended transition function



- Language of an ε-NFA is the set of strings w such that δ_E(q_0,w) contains a final state.

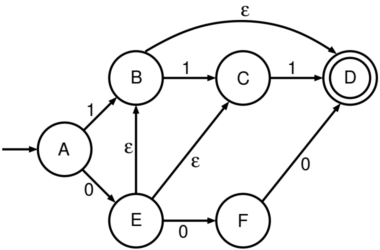
Equivalence of NFA, ε-NFA

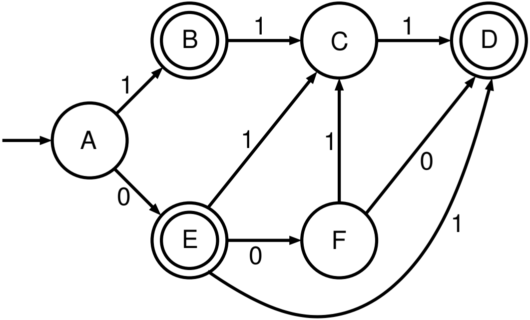
- Every NFA is an ε-NFA.
  - It just has no ε-transitions.
- Converse requires us to take an ε-NFA and construct an NFA that accepts the same language.
- This is done by combining ε-transitions with the next transition on a real input.
- Start with an ε-NFA (Q,Σ,q_0,F,δ_E) and construct an ordinary NFA (Q,Σ,q_0,F',δ_N).

Equivalence of NFA, ε-NFA

- Compute δ_N(q,a) as follows:
  - Let S = CL(q).
  - δ_N(q,a) is the union over all p in S of δ_E(p,a).
- F' = set of states q such that CL(q) contains a state of F.
- **Intuition:** δ_N incorporates ε-transitions before using a.
- Proof of equivalence is by induction on |w| that CL(δ_N(q_0,w)) = δ_E(q_0,w).
- **Basis:** CL(δ_N(q_0,ε)) = CL(q_0) = δ_E(q_0,ε).
- **Inductive step:** Assume **IH** is true for all x shorter than w. Let w = xa.
  - Then CL(δ_N(q_0,xa)) = CL(δ_E(CL(δ_N(q_0,x)),a)) (by definition).
  - But from **IH**, CL(δ_N(q_0,x)) = δ_E(q_0,x).
  - Hence, CL(δ_N(q_0,w)) = CL(δ_E(δ_E(q_0,x),a)) = δ_E(q_0,w).

Example





Summary

- DFA's, NFA's and ε-NFA's all accept exactly the same set of languages: the regular languages.
- NFA types are easier to design and may have exponentially fewer states than a DFA.
- But only a DFA can be implemented!