CSE 2001:
Introduction to Theory of Computation

Winter 2013

Suprakash Datta

datta@cse.yorku.ca

Office: CSEB 3043

Phone: 416-736-2100 ext 77875

Course page: http://www.cse.yorku.ca/course/2001

Recall: Regular Languages

The language recognized by a finite automaton M is denoted by **L(M).**

A regular language is a language for which there exists a recognizing finite automaton.

Terminology: closure

- A set is defined to be closed under an operation if that operation on members of the set always produces a member of the same set. (adapted from wikipedia)

E.g.:

- The integers are closed under addition, multiplication.
- The integers are not closed under division
- Σ^* is closed under concatenation

- A set can be defined by closure -- Σ^* is called the (Kleene) closure of Σ under concatenation.

Terminology: Regular Operations

Pages 44-47 (Sipser)

The regular operations are:

1. Union
2. Concatenation
3. Star (Kleene Closure): For a language A,

   $A^* = \{w_1 w_2 w_3 \ldots w_k \mid k \geq 0, \text{ and each } w_i \in A\}$

Closure Properties

- Set of regular languages is closed under
  - -- Complementation
  - – Union
  - – Concatenation
  - – Star (Kleene Closure)

Complement of a regular language

- Swap the accepting and non-accept states of M to get M’.

- The complement of a regular language is regular.

Other closure properties

Union: Can be done with DFA, but using a complicated construction.

Concatenation: We tried and failed

Star: ???

We introduced non-determinism in FA

Recall: NFA drawing conventions

- Not all transitions are labeled
- Unlabeled transitions are assumed to go to a reject state from which the automaton cannot escape

Closure under regular operations

Union (new proof):

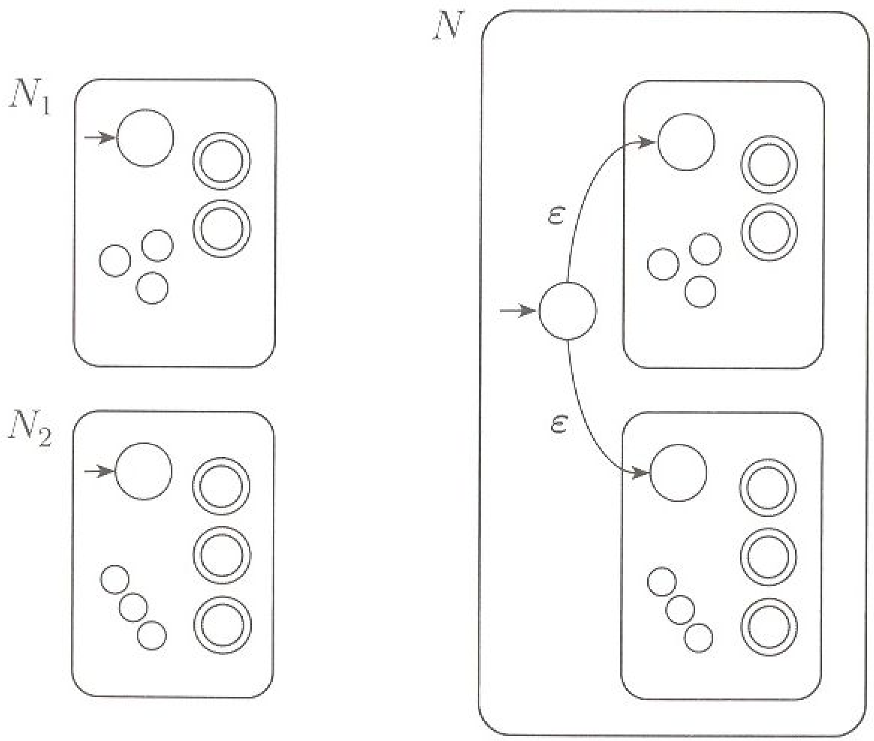

FIGURE 1.46
Construction of an NFA N to recognize $A_1 \cup A_2$

Closure under regular operations

Concatenation:

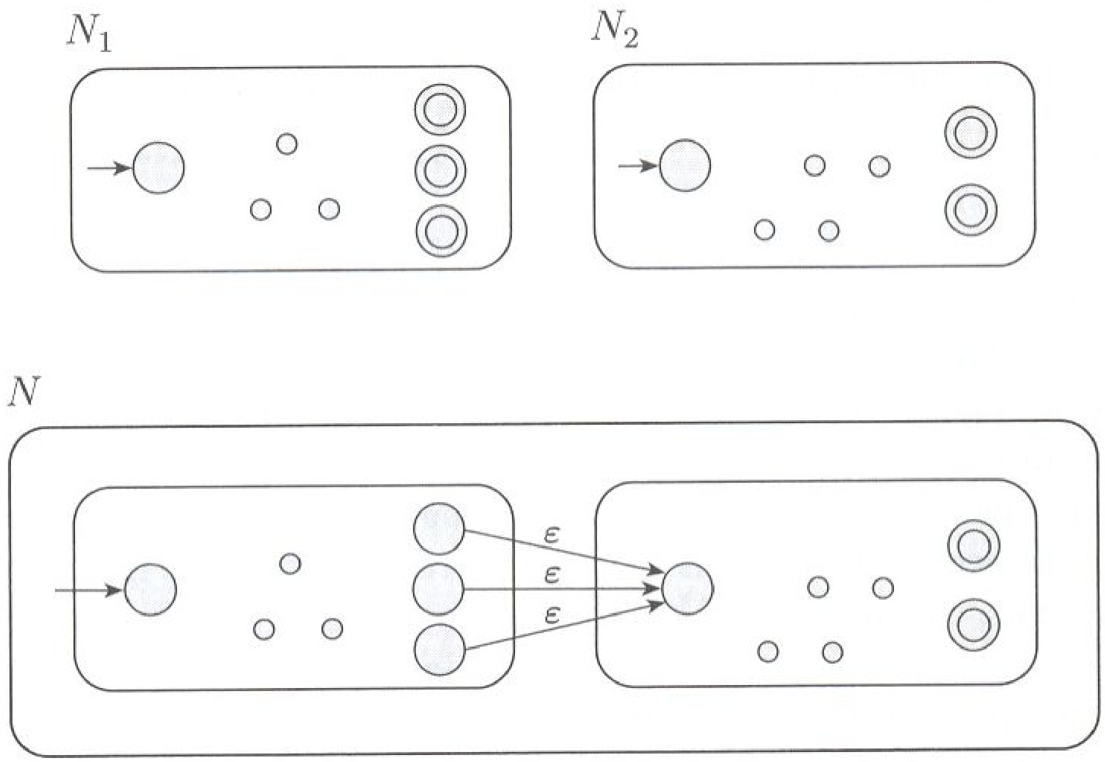

FIGURE 1.48
Construction of N to recognize $A_1 \circ A_2$

Closure under regular operations

Star:

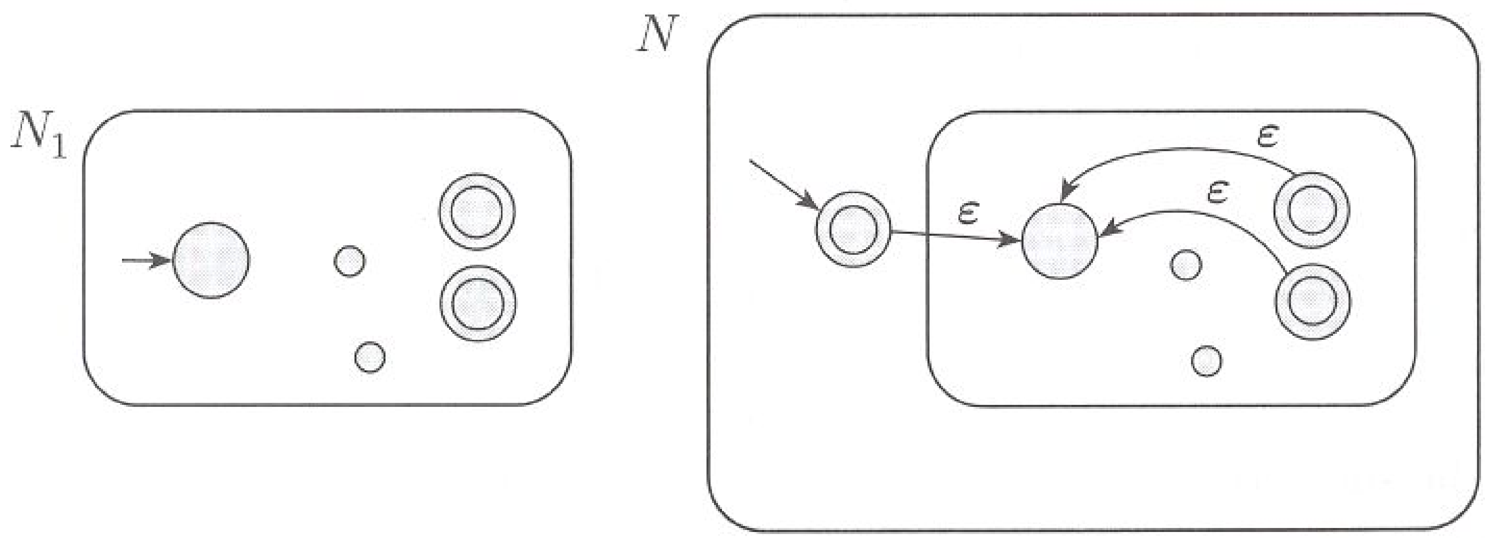

FIGURE 1.50
Construction of N to recognize A^*

Incorrect reasoning about RL

- Since $L_1 = \{w \mid w=a^n, n \in \mathbf{N}\}$, $L_2 = \{w \mid w = b^n, n \in \mathbf{N}\}$ are regular, therefore $L_1 \bullet L_2 = \{w \mid w=a^n b^n, n \in \mathbf{N}\}$ is regular
- If L_1 is a regular language, then $L_2 = \{w^R \mid w \in L_1\}$ is regular, and Therefore $L_1 \bullet L_2 = \{w\, w^R \mid w \in L_1\}$ is regular

Are NFA more powerful than DFA?

- NFA can solve every problem that DFA can (DFA are also NFA)
- Can DFA solve every problem that NFA can?

Equivalence of NFA, DFA

- Pages 54-58 (Sipser, 2nd ed)
- We will prove that every NFA is equivalent to a DFA (with upto exponentially more states).
- Non-determinism does not help FA's to recognize more languages!

Epsilon Closure

- Let $N=(Q,\Sigma,\delta,q_0,F)$ be any NFA
- Consider any set $R \subseteq Q$
- $E(R) = \{q | q$ can be reached from a state in R by following 0 or more ε-transitions$\}$
- $E(R)$ is the epsilon closure of R under ε-transitions

Proving equivalence

For all languages $L \subseteq \Sigma^*$

$$L = L(N) \quad \text{iff} \quad L = L(M)$$

for some NFA N for some DFA M

One direction is easy:

A DFA M is also a NFA N. So N does not have to be `constructed' from M

Proving equivalence – contd.

The other direction:

Construct M from N

- $N = (Q,\Sigma,\delta,q_0,F)$
- Construct $M = (Q',\Sigma,\delta',q'_0,F')$ such that,
  - for any string $w \in \Sigma^*$,
  - w is accepted by N iff w is accepted by M

Special case

- Assume that ε is not used in the NFA N.
  - Need to keep track of each subset of N
  - So $Q' = \mathcal{P}(Q)$, $q'_0 = \{q_0\}$
  - $\delta'(R,a) = \cup(\delta(r,a))$ over all $r \in R$, $R \in Q'$
  - $F' = \{R \in Q' \mid R$ contains an accept state of N$\}$

- Now let us assume that ε is used.

Construction (general case)

1. $Q' = \mathcal{P}(Q)$
2. $q'_0 = E(\{q_0\})$
3. for all $R \in Q'$ and $a \in \Sigma$
   $\delta'(R, a) = \{q \in Q | q \in E(\delta(r,a))$ for some $r \in R\}$
4. $F' = \{ R \in Q' |$ R contains an accept state of N$\}$

Why the construction works

- for any string w $\in \Sigma^*$,
- w is accepted by N iff w is accepted by M
- Can prove using induction on the number of steps of computation…

State minimization

It may be possible to design DFA’s without the exponential blowup in the number of states. Consider the NFA and DFA below.