Lec 02. Nondeterministic Finite Automata

Eunjung Kim

Nondeterminism

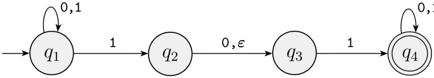


Figure 1.27, Sipser 2012.

	Deterministic FA	Nondeterministic FA
each state & symbol	one leaving arc	multiple arcs or none
labels	Σ	$\Sigma \cup \{\epsilon\}$
computation history	single path	multiple paths (tree)

NONDETERMINISM: COMPUTATION TREE AND ϵ

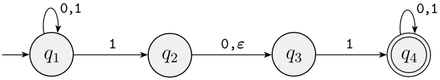
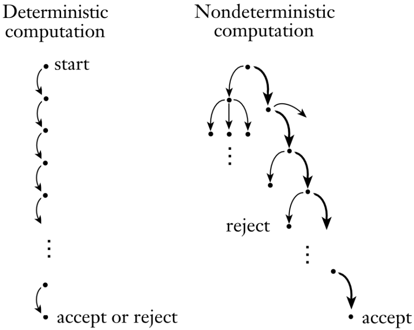


Figure 1.27, Sipser 2012.



Examples of NFA

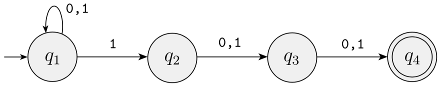


Figure 1.31, Sipser 2012.

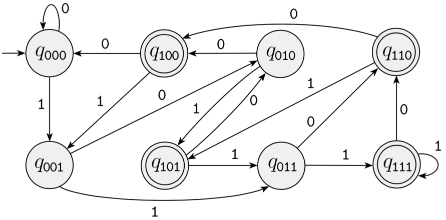


Figure 1.32, Sipser 2012.

Examples of NFA

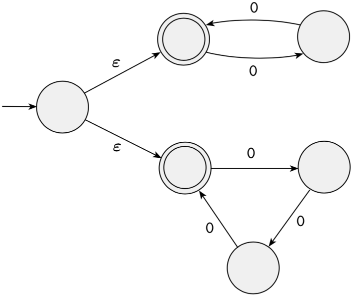


Figure 1.33, Sipser 2012.

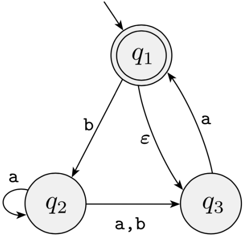
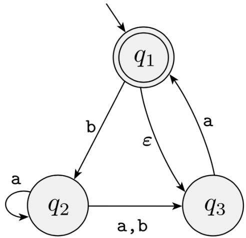


Figure 1.36, Sipser 2012.

Formal definition of NFA

Nondeterministic FA is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$

- Q a finite set called the states,
- Σ a finite set called the alphabet,
- δ a function from $Q \times \Sigma_\epsilon$ to 2^Q called the transition function,
- $q_0 \in Q$ the start state,
- $F \subseteq Q$ the set of accept states.

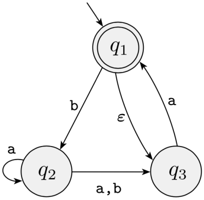


Write a formal description of this NFA

Language recognized by NFA

NFA N accepts w if

1. w can be written as $y_1, \ldots, y_m$ with $y_i \in \Sigma_\epsilon = \Sigma \cup \{\epsilon\}$,
2. there exists a sequence of states $r_0, \ldots, r_m$ s.t.
   - $r_0 = q_0$,
   - $r_{i+1} \quad (??) \quad \delta(q_i, y_i)$,
   - $r_m \in F$.



Write a computation tree for $w = baabaaa$. How many accepting paths?

Closure under regular operation

Union operation

Let A_1 and A_2 be two languages recognized by NFA N_1 and N_2 respectively. Then $A_1 \cup A_2$ is recognized by some NFA.

Closure under regular operation

Concatenation operation

Let A_1 and A_2 be two languages recognized by NFA N_1 and N_2 respectively. Then $A_1 \circ A_2$ is recognized by some NFA.

Closure under regular operation

Kleene star operation

Let A a languages recognized by NFA N. Then A^* is recognized by some NFA.

Closure under complementation

Complementation operation

Let A a languages recognized by NFA N. Then $\bar{A}$, that is, $\Sigma^* - A$ is recognized by some NFA.

- For a regular language L, we can obtain a DFA recognizing the complement of L.
- ...using the trick...
- Can we use the same trick for NFA in general?

Closure under intersection

Intersection operation

Let A_1 and A_2 be two languages recognized by NFA N_1 and N_2 respectively. Then $A_1 \cap A_2$ is recognized by some NFA.

- Use the expression that $A_1 \cap A_2 =$??????.
- Combine the above (which ones?) operations on NFAs...
- Direct way with two DFAs M_1 and M_2 by <u>simulating</u> both automata <u>simultaneously</u>.

Closure under intersection

- Direct way with two DFAs M_1 and M_2 by simulating both automata simultaneously.

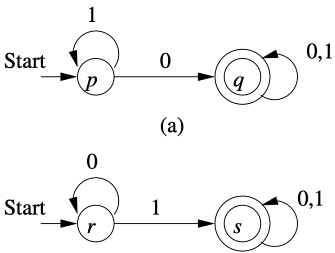


Figure 4.4 (a)-(b), Hopcroft et al. 2014.

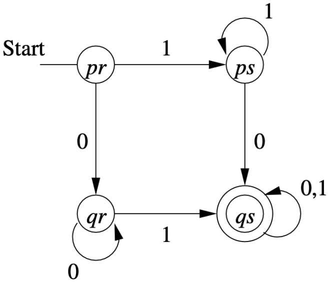


Figure 4.4 (c), Hopcroft et al. 2014.

Equivalence of NFA and DFA

NFA and DFA own the same computational power

For every NFA, there exists a deterministic finite automata which recognizes the same language (a.k.a. equivalent DFA).

Proof outline.

- Let $N = (Q, \Sigma_\epsilon, \delta, q_0, F)$ be an NFA.
- We want to construct a DFA $M = (Q', \Sigma, \delta', q_0', F')$ such that $L(N) = L(M)$.
- Define $Q' := 2^Q$, i.e. the collection of all subsets of Q.
- Let us define δ', $q_0' \in 2^Q$ and $F' \subseteq 2^Q$,

Proof: constructing DFA M, without ϵ-transition

- $q'_0 = q_0$.
- transition function δ' from $2^Q \times \Sigma$ to 2^Q:
  for every $R \in 2^Q$ (R is a subset of Q) and every symbol $a \in \Sigma$,

$$\delta'(R, a) := \bigcup_{r \in R} \delta(r, a)$$

- Define $F' \subseteq 2^Q$ as the collection of all subsets of Q containing at least one accept state of N.

Proof: constructing DFA M

- How to define the initial state for DFA: from a state $q \in Q$ of NFA N, any other state q' that can be reached by reading a string ϵ, can be aggregated with q to form a single state in DFA.
- Define $ext(q) \subseteq Q$ as the set of all states q' of N such that there is a directed path from q to q' in (the state diagram of) N each of whose arcs carries the label ϵ. Extend the definition $ext(X) := \bigcup_{q \in X} ext(q)$.
- transition function δ' from $2^Q \times \Sigma$ to 2^Q:
  for every $R \in 2^Q$ (R is a subset of Q) and every symbol $a \in \Sigma$,

$$\delta'(R, a) := ext\Big(\bigcup_{r \in R} \delta(r, a)\Big)$$

- Define $q'_0 := ext(q_0) \in 2^Q$. Note that q'_0 corresponds to a subset of Q.
- Define $F' \subseteq 2^Q$ as the family of all subsets of Q containing at least one accept state of N.

Proof: $L(N) \subseteq L(M)$

- Let $\pi = (q_0, w = w_0), \ldots, (q_i, w_i), \ldots, (q_s, w_s = \epsilon)$ be an accepting computation history of N for w, that is

$$q_i \in \delta(q_{i-1}, y_i) \text{ for every } i \in [s] \text{ and } q_s \in F,$$

  where $w_i = y_i \circ w_{i-1}$ with $y_i \in \Sigma \cup \{\epsilon\}$ for every $i \in [s]$.

- Let $0 \leq i_0 < \cdots < i_{t-1} < s$ be the indices in π such that the leading symbol of w_i in (q_i, w_i) is in Σ (not ϵ). Let $i_t = s$.

- Let $Q_0 = q_0'$. Inductively for each $i \in [t]$, let

$$Q_i = \delta'(Q_{i-1}, y_i).$$

- Now we have a computation history for $w = y_1 \cdots y_t$ as follows

$$\pi' = (Q_0, w), (Q_1, y_2 \cdots y_t), \cdots (Q_t, \epsilon).$$

Proof: $L(N) \subseteq L(M)$

- It remains to see Q_t is an accept state of M, i.e. the subset $Q_t \subseteq Q$ contains at least one accept state of N.
- It suffices to prove $q_{i_j} \in Q_j$ for each $0 \leq j \leq t$ (this implies the above).
- By induction. Base case holds because $Q_0 = q'_0 = ext(q_0)$ and q_{i_0} is reachable from q_0 by reading a number of ϵ's.
- From the partition using $i_0, \ldots, i_t$, q_{i_j} is reachable from $q_{i_{j-1}}$ by reading a single symbol $a_{i_{j-1}+1} \in \Sigma$ and additional ϵ's. Hence $q_{i_j} \in ext(\delta(q_{i_{j-1}}, a_{i_{j-1}+1}) \subseteq ext(\delta(Q_{j-1}, a_{i_{j-1}+1}) = \delta'(Q_{j-1}, a_{i_{j-1}+1}) = Q_j$.

Proof: $L(M) \subseteq L(N)$

- Let $\pi' = (Q_0, w = w_0), \ldots, (Q_i, w_i), \ldots, (Q_t, w_t = \epsilon)$ be an accepting computation history of M for w. By definition of computation history $Q_i = \delta'(Q_{i-1}, x_i)$, where $x_i \in \Sigma$ is the leading symbol of w_{i-1}.
- We construct an accepting computation history of N by following the sequence π' backwardly.
- Let $q_f \in Q_t$ be an accept state of N.
- Observe: for each $q \in Q_i \subseteq Q$, there exists a state $q' \in Q_{i-1}$ such that from q' to q there is a computation history consisting of readying a string consisting of x_i followed by ϵ's.
- Now starting from q_f, we concatenate computation histories witnessed by the previous observation.

Proof: constructing DFA M with ϵ-transition

- Define $ext(q) \subseteq Q$ as the set of all states q' of NFA N such that there is a directed path from q to q' in (the state diagram of) N each of whose arcs carries the label ϵ. Let $ext(X) := \bigcup_{q \in X} ext(q)$.

- transition function δ' from $2^Q \times \Sigma_\epsilon$ to 2^Q:
  for every $R \in 2^Q$ (R is a subset of Q) and every symbol $a \in \Sigma$,

$$\delta'(R, a) := ext(\bigcup_{r \in R} \delta(r, a))$$

- Define $q'_0 := ext(q_0) \in 2^Q$. Note that q'_0 corresponds to a subset of Q.
- Define $F' \subseteq 2^Q$ as the family of all subsets of Q containing at least one accept state of N.