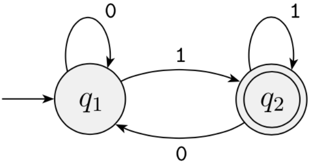
Lec 02. More on DFA & Nondeterministic Finite Automata

Eunjung Kim

Formal definition of computation

- Let $w = w_1 w_2 \cdots w_n \in \Sigma^*$, where $w_i \in \Sigma$.
- The extended transition function $\hat{\delta}$ is a mapping from $Q \times \Sigma^*$ to Q defined as: $\hat{\delta}(q, w) = q'$ if there is a sequence of states $r_0, \ldots, r_n$ in Q such that
  - $r_0 = q$,
  - $r_i = \delta(r_{i-1}, w_i)$ for every $1 \leq i \leq n$,
  - $r_n = q'$
- Equivalently, there is a walk in the transition diagram of M from q to q' labelled by w.

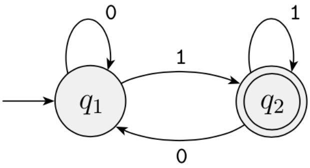


Computation history

- Configuration of a finite automata $M = (Q, \Sigma, \delta, q_0, F)$ is a pair $(q, w) \in Q \times \Sigma^*$.
- We interpret a configuration (q, w) as...
- $(q, w) \rightsquigarrow_M (q', w')$ if...
- $(q, w) \rightsquigarrow_M^* (q', w')$ if there is...
- A sequence of configuration is a computation history if the first configuration is in the form (q_0, w) for some $w \in \Sigma^*$, and each contiguous configurations are related by $\rightsquigarrow_M$ or $\rightsquigarrow_M^*$.
- A computation history is an accepting computation history if the last configuration is in the form ??????.

DFA M ACCEPTS A STRING

- Let $w_1 w_2 \cdots w_n$ be a string in Σ^* with $w_i \in \Sigma$ for each i.
- $M = (Q, \Sigma, \delta, q_0, F)$ accepts w if
  - $\hat{\delta}(q_0, w) \in F$, or equivalently
  - In the transition diagram of M, there is an walk from q_0 to an accept state labelled by w.



Language recognized by DFA

Definition: language recognized by DFA

- Let $M = (Q, \Sigma, \delta, q_0, F)$ be a finite automata.
- A string $w \in \Sigma^*$ is accepted by M if
  - $\hat{\delta}(q_0, w) \in F$, or equivalently
  - in the transition diagram of M, there is an walk from q_0 to an accept state labelled by w.
- Let $L(M)$ be the set of all strings which are accepted by M.
- A language A is said to be recognized by M if $A = L(M)$.

Regular Language

Regular language = recognized by some DFA

- A language L over a finite alphabet is said to be regular if there is a (deterministic) finite-state automaton M which recognizes L.

FROM LANGUAGES TO DFA: EXAMPLES

SHOW THAT THE FOLLOWING LANGUAGE IS REGULAR.

- $L = \{$all 0,1-strings containing 01$\}$
- $L = \{$ all 0,1-strings containing exactly even numbers of 0's and 1's respectively $\}$.
- $L = \{$ all strings containing at least two a's$\} \subseteq \{a, b\}^*$.
- $L = \{awa : w \in \{a, b\}^*\}$.

From Languages to DFA: examples

Suppose $L \subseteq \Sigma^*$ is regular. Is the complement of L, i.e. $\Sigma^* - L$, is regular?

From Languages to DFA: examples

$L = \{awa : w \in \{a,b\}^*\}$, $L^2 = \{aw_1aaw_2a : w_i \in \{a,b\}^*\}$

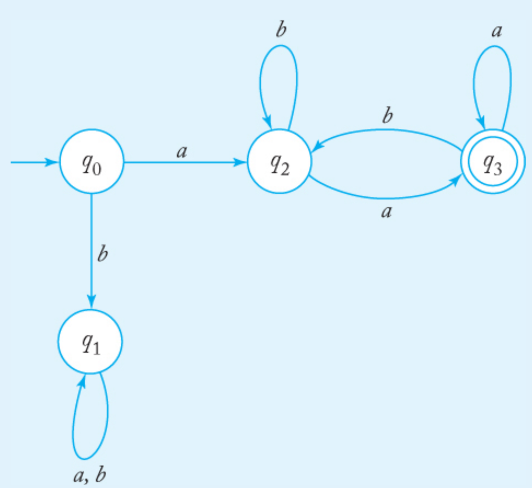


Figure 2.6 from Linz 2017.

Nondeterminism

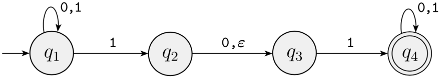


Figure 1.27, Sipser 2012.

	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

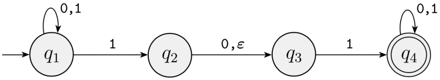
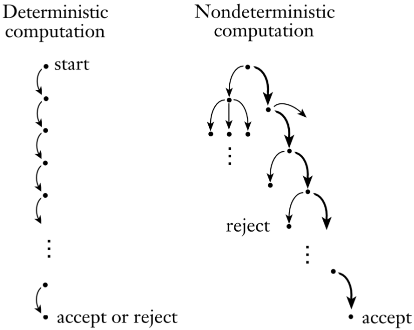


Figure 1.27, Sipser 2012.



Examples of NFA

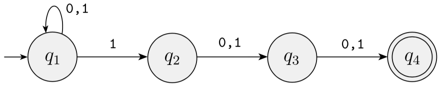


Figure 1.31, Sipser 2012.

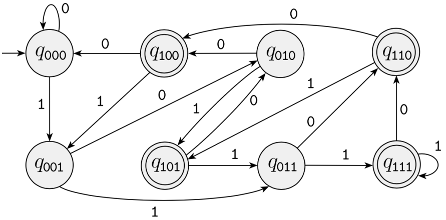


Figure 1.32, Sipser 2012.

Examples of NFA

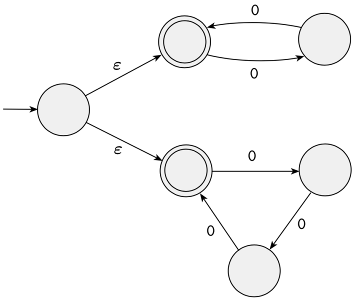


Figure 1.33, Sipser 2012.

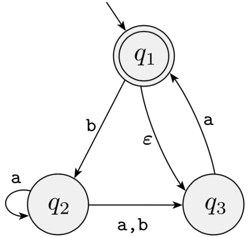
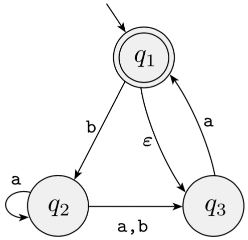


Figure 1.36, Sipser 2012.

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.

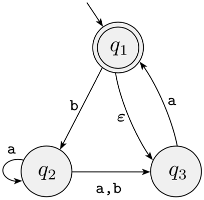


Write a formal description of this NFA

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 simulating both automata simultaneously.

Closure under intersection

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

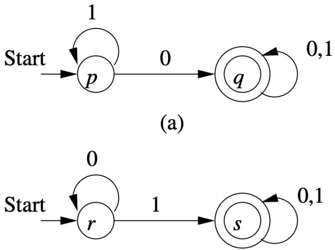


Figure 4.4 (a)-(b), Hopcroft et al. 2014.

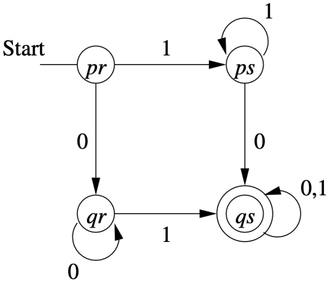


Figure 4.4 (c), Hopcroft et al. 2014.