CS310 : Automata Theory 2019

Lecture 3: Nondeterministic finite automaton (NFA)

Instructor: Ashutosh Gupta

IITB, India

Compile date: 2019-01-08

Announcements

- Course webpage
  `http://www.cse.iitb.ac.in/~akg/courses/2019-cs310/`
- Optional tutorials (Tuesdays 7PM?)
- Join piazza (invites in couple of hours; all communications via piazza)
- First quiz on 23rd, 8:30AM
- Due to logistical limits, we can not accept non-CSE students. Plenty of electives to choose from.

What we know?

- Problems are languages
- Deterministic finite automaton (DFA)
- DFAs recognize regular languages

Topic 3.1

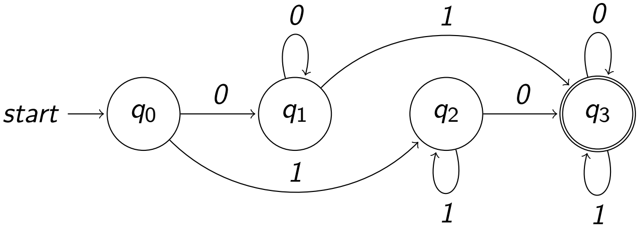
Example regular languages

Example: DFAs for recognizing languages

Example 3.1

Let $\Sigma = \{0, 1\}$. *Consider language* $\{w | \textit{both } 0 \textit{ and } 1 \textit{ occur in } w\}$
We choose four states

- q_0 *interpretation "nothing matched yet"*
- q_1 *interpretation "seen 0"*
- q_2 *interpretation "seen 1"*
- q_3 *interpretation "seen both"*

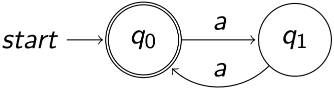


Example: DFAs for recognizing languages

Example 3.2

Let $\Sigma = \{a\}$. Consider language

$$L(A_1) = \{w \mid \ |w|\ mod\ 2 = 0\}$$



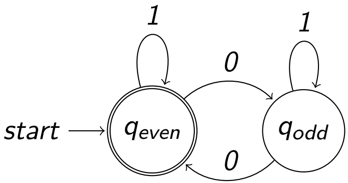
Exercise 3.1

Give a language that only accepts words of length multiples of three.

Example: DFAs for recognizing languages

Example 3.3

Let $\Sigma = \{0, 1\}$. Let $L = \{w | w$ has even number of 0s$\}$



The above DFA recognizes L.

Exercise 3.2

Give DFAs for the following languages

- $\{w | w$ *has even number of 0s and odd number of 1s*$\}$
- $\{w |$ *01 occurs in w somewhere*$\}$
- $\{w |$ *0s in w are multiples of three apart*$\}$

Topic 3.2

Nondeterminism

Modelling unknowns

For many systems under study, we may have

- unknown inputs,
- unknown internal parameters, or
- too complex components to model.

How do we model them in our study of computing?

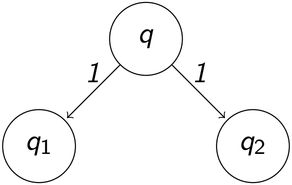
Nondeterminism

We call the situation nondeterminism. We model it as follows.

Upon reading an input symbol, an automaton can move to one of multiple states.

Example 3.4

Upon reading 1*, the automaton can jump from* q *to either* q_1 *or* q_2*.*



Exercise 3.3

Do we have free will?

Example: nondeterminism in action

At first nondeterminism is a difficult to comprehend concept!

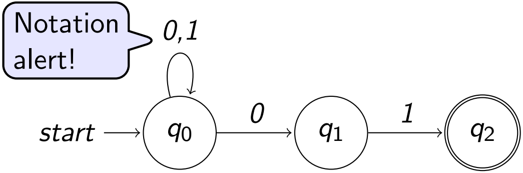
However, a few examples should make it clear.

Example 3.5

Let us suppose we want an automaton that recognizes the following language.

$$\{w \mid w \text{ ends with } 01\}$$

Consider the following nondeterministic automaton for recognizing it



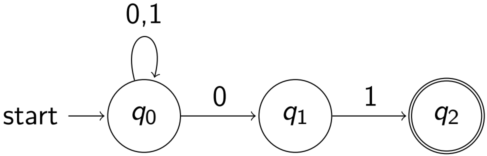
Notation alert!

- At q_0, if 0 is read there are two possible available transitions.
- At q_1 there is no transition for input 0 and q_2 has no outgoing transitions

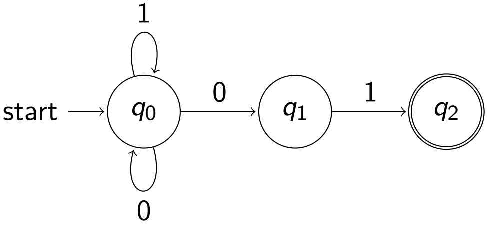
Notation alert: multiple labels on transition

Multiple input symbols on transitions are the natural extension of the notation.

Example 3.6



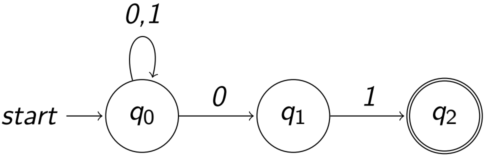
The above is denoting the following.



Nondeterminism causes multiple runs

There can be several or no runs for a given input word.

Example 3.7



Let 10101 be an input word. The following are potential runs.

- $q_0 \xrightarrow{1} q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{0}$ *stuck* unfinished runs are unacceptable
- $q_0 \xrightarrow{1} q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_2$ accepted run

If the word has 01 at the end, there exists an accepting run!

Recall, DFA for the language was lot more involved! The above is concise.

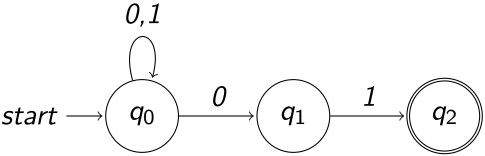
Exercise 3.4

Give runs for word 1110.

Power of guess

Nondeterminism brings the "power of guess" to DFA.

Example 3.8



The automaton has to guess when to leave q_0. If the word is acceptable and the guess is right, it reaches the accepting state.

Nondeterministic finite automaton (NFA)

Definition 3.1

A nondeterministic finite automaton(NFA) A is a five-tuple

$$(Q, \Sigma, \delta, q_0, F)$$

where

- Q *is a finite set of states,*
- Σ *is a finite set of input symbols,*
- $\delta : Q \times \Sigma \rightarrow \mathfrak{p}(Q)$ *is a function that takes a state and an input symbol as input and returns the set of possible next states,*
- $q_0 \in Q$ *is the start/initial state, and*
- $F \subseteq Q$ *is a set of accepting states.*

Exercise 3.5

How this definition models stuck executions?

Commentary: Only the definition of δ is different from the DFA.

Notation alert : power set

Definition 3.2

$\mathfrak{p}(Q)$ *is the set of all subsets of* Q*.*

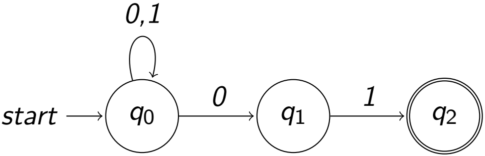
Example 3.9

Let $Q = \{q_0, q_1, q_2\}$*.*

$$\mathfrak{p}(Q) = \{\emptyset, \{q_0\}, \{q_1\}, \{q_2\}, \{q_0, q_1\}, \{q_1, q_2\}, \{q_2, q_0\}, \{q_0, q_1, q_2\}\}$$

Example NFA

Example 3.10



We write the above automaton according to the formal definition as follows.

$A = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\})$, *where* δ *is the following*

States

Inputs

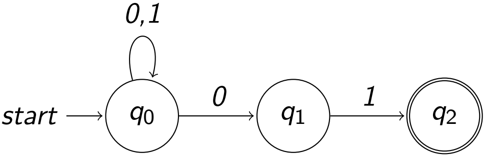
δ	q_0	q_1	q_2
0	$\{q_0, q_1\}$	$\emptyset$	$\emptyset$
1	$\{q_0\}$	$\{q_2\}$	$\emptyset$

Run of an NFA

Definition 3.3

Let $A = (Q, \Sigma, \delta, q_0, F)$ be an NFA. A run of A on a word $a_1, \dots, a_n$ is a sequence of states $q_0, \dots, q_n$ such that $q_i \in \delta(q_{i-1}, a_i)$ for each $1 \leq i \leq n$.

Example 3.11



Consider word $w = 0101$,

a run of A on w is $q_0 q_0 q_0 q_1 q_2$.

The runs are required to be finished! Stuck runs are not counted.

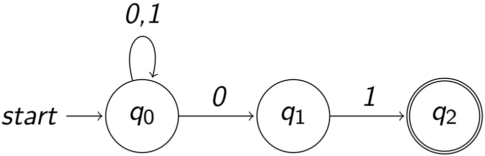
Extending the transition function to words

Definition 3.4

Let $A = (Q, \Sigma, \delta, q_0, F)$ be an NFA. Let $\hat{\delta} : Q \times \Sigma^* \rightarrow \mathfrak{p}(Q)$ be defined as follows.

$$\hat{\delta}(q, \epsilon) \triangleq \{q\}$$
$$\hat{\delta}(q, wa) \triangleq \bigcup_{q' \in \hat{\delta}(q,w)} \delta(q', a)$$

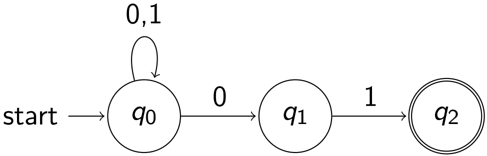
Example 3.12



$\hat{\delta}(q_0, 0) = \{q_0, q_1\}$

$\hat{\delta}(q_0, 01) = \bigcup_{q' \in \hat{\delta}(q_0,0)} \hat{\delta}(q', 1) = \hat{\delta}(q_0, 1) \cup \hat{\delta}(q_1, 1) = \{q_0, q_2\}$

Exercise: extended transitions



Exercise 3.6

Give value of the following function applications

- $\hat{\delta}(q_2, 01) =$
- $\hat{\delta}(q_1, 01) =$
- $\hat{\delta}(q_0, 00) =$
- $\hat{\delta}(q_0, 110) =$

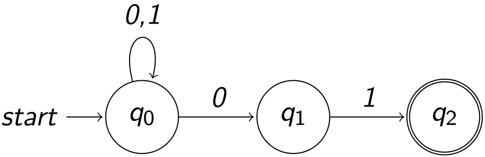
Accepted word

Definition 3.5

A word w is accepted by an NFA $A = (Q, \Sigma, \delta, q_0, F)$ if $\hat{\delta}(q_0, w) \cap F \neq \emptyset$.

Example 3.13

Consider the following NFA



Since $\hat{\delta}(q_0, 1101) = \{q_0, q_2\}$, 1101 is accepted by the above NFA.

Language of an NFA

Definition 3.6

The language of an NFA $A = (Q, \Sigma, \delta, q_0, F)$ is the set of words that are accepted by A. We denote the language by $L(A)$. In set notation,

$$L(A) = \{w|\hat{\delta}(q_0, w) \cap F \neq \emptyset\}.$$

We also say that A recognizes language $L(A)$.

A DFA is also an NFA

Theorem 3.1

For a DFA $A = (Q, \Sigma, \delta, q_0, F)$, there is an NFA $A' = (Q, \Sigma, \delta', q_0, F)$ such that $L(A) = L(A')$.

Proof.

All parts of A' are already defined except δ'.

We construct δ' as follows

$$\delta'(q, a) \triangleq \{\delta(q, a)\}.$$

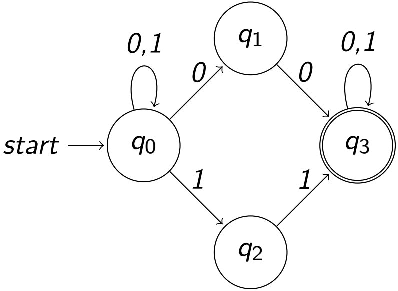
It is trivial to see that A and A' recognize same languages. □

Example: modeling 'or' using NFA

Example 3.14

Let $\Sigma = \{0, 1\}$*. Consider language*

$$\{w | either\ 00\ or\ 11\ occur\ w\}$$



More languages by NFA

With the power of guess, do NFA recognize more languages?

To be continued...

End of Lecture 3