CS310 : Automata Theory 2019

Lecture 2: Deterministic finite automaton (DFA)

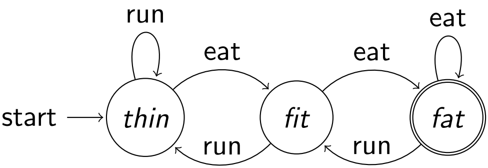
Instructor: Ashutosh Gupta

IITB, India

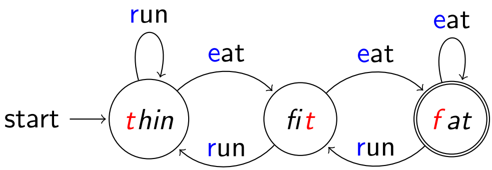
Compile date: 2019-01-07

What we know?

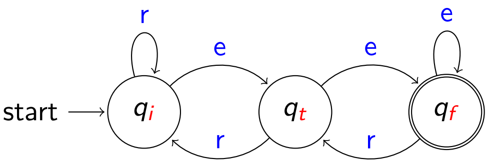
- Alphabet Σ
- Words from Σ^*
- States Q
- Transitions
- Accepting states



Better names for our running example



Let us use shorter symbols for our running example.



Topic 2.1

Deterministic finite automaton

At every step exactly one move possible!

Deterministic finite automaton

Q is finite

Now let us see the formal definition and learn to read math!

Commentary: Mouthful name! It should have been simply called "basic automaton"!

Deterministic finite automaton

Definition 2.1

A deterministic finite automaton(DFA) 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,*
- $\delta : Q \times \Sigma \to Q$ *is a function that takes a state and an input symbol as input and returns the next state,*
- $q_0 \in Q$ *is the start/initial state, and*
- $F \subseteq Q$ *is a set of accepting states.*

representing the transition graph

Exercise 2.1

a. *Can* Q *be empty?*

b. *Can* Σ *be empty?*

c. *Can* F *be empty?*

Notation alert: declaring and representing functions

The following notation declares a function f that takes N inputs of various types and returns output of type $OutputType$.

$$f : Input\_1\_Type \times \cdots \times Input\_N\_Type \rightarrow OutputType$$

If all types are finite, the functions may be called maps.

Notation alert: maps as tables

A map can be give to us as a table.

Example 2.1

and : $\mathcal{B} \times \mathcal{B} \rightarrow \mathcal{B}$ *can be defined as follows.*

and	0	1
0	0	0
1	0	1

Input 1

Input 2

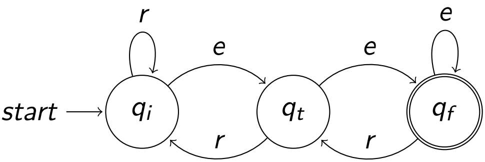
Output values

We can also write the function in the following notation

- $and(0, 0) = 0$
- $and(0, 1) = 0$
- $and(1, 0) = 0$
- $and(1, 1) = 1$

Example: deterministic finite automaton

Example 2.2



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

$A = (\{q_i, q_t, q_f\}, \{r, e\}, \delta, q_i, \{q_f\})$, *where* δ *is the following table*

States / *Inputs*

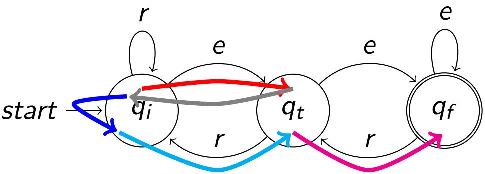
δ	q_i	q_t	q_f
e	q_t	q_f	q_f
r	q_i	q_i	q_t

Run of automaton

Definition 2.2

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

Example 2.3



Consider word $w = erree$

Run on the word $q_i q_t q_i q_i q_t q_f$

Commentary: Length of run is one longer than the input word

Extending the transition function to words

Definition 2.3

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

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

More general notion than "run" (how?)

Example 2.4

Consider transition function

δ	q_i	q_t	q_f
e	q_t	q_f	q_f
r	q_i	q_i	q_t

$$\hat{\delta}(q_t, eer) = \delta(\hat{\delta}(q_t, ee), r) = \delta(\delta(\hat{\delta}(q_t, e), e), r) = \delta(\delta(\delta(\hat{\delta}(q_t, \epsilon), e), e), r)$$
$$= \delta(\delta(\delta(q_t, e), e), r) = \delta(\delta(q_f, e), r) = \delta(q_f, r) = q_t$$

Exercise 2.2

Give value of the following function applications

- $\hat{\delta}(q_f, eer) =$
- $\hat{\delta}(q_i, eer) =$
- $\hat{\delta}(q_f, rr) =$
- $\hat{\delta}(q_i, rree) =$

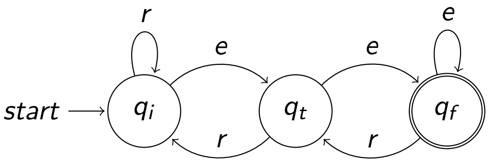
Accepted word

Definition 2.4

A word w is accepted by a DFA $A = (Q, \Sigma, \delta, q_0, F)$ if $\hat{\delta}(q_0, w) \in F$.

Example 2.5

Consider the following DFA



Since $\hat{\delta}(q_i, rree) = q_f$, rree is accepted by the above DFA.

Language of a DFA

Definition 2.5

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

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

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

Definition 2.6

A language L is a regular language if there is a DFA A such that $L = L(A)$.

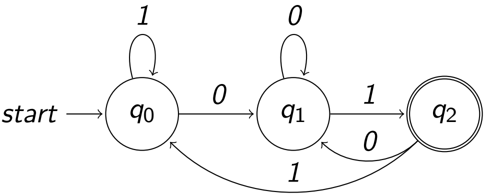
Example: DFA recognizing languages

Example 2.6

Let $L = \{w | w \text{ ends with } 01\}$

We choose three states

- q_0 *interpretation "nothing matched yet"*
- q_1 *interpretation "recently seen 0"*
- q_2 *interpretation "recently seen 01"*



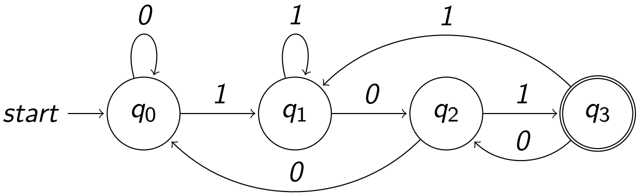
Example: DFA recognizing languages

Example 2.7

Let $L = \{w | w \text{ ends with } 101\}$

We choose three states

- q_0 *interpretation "nothing matched yet"*
- q_1 *interpretation "recently seen 1"*
- q_2 *interpretation "recently seen 10"*
- q_3 *interpretation "recently seen 101"*



First non-trivial question

Are there languages that are not regular?

If yes, how do we recognize they are regular or not?

To be continued...

End of Lecture 2