Lecture #2: Introduction to Deterministic Finite Automata
What Will Happen During the Lecture

Remember... You Had Homework!

Students were asked to work through the following set of lecture notes before this lecture.

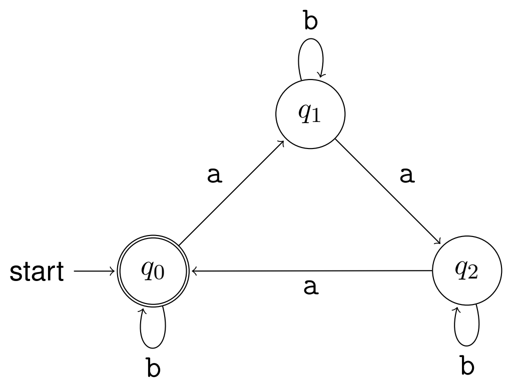
- Lecture Notes — “Introduction to Deterministic Finite Automata”.

If possible, students should also complete the “Mathematics Review” before this lecture.

Once again, you may attend the lecture presentation if you have not worked through this material ahead of time — but it will not be repeated for you, and you might get a little bit lost, during the presentation, if you haven’t worked through this.

Consideration of a Deterministic Finite Automaton

We will consider a deterministic finite automaton M that has alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ and that can be represented as follows.



This will used to consider how to describe a deterministic finite automaton in more than one way, and how to decide whether a given string is accepted by a given deterministic finite automaton.

It will also be used to describe how one can sometimes ***prove*** that a given deterministic finite automaton has a given language.

Establishing and Using Properties of Deterministic Finite Automata

Consider a deterministic finite automaton $M = (Q, \Sigma, \delta, q_0, F)$ — and recall that the ***transition function*** $\delta : Q \times \Sigma \to Q$ is a ***total function***. Another function, called the ***extended transition function***

$$\delta^\star : Q \times \Sigma^\star \to Q$$

was defined in ***two ways***, for a given state $q \in Q$ and for a given string

$$\omega = a_1 a_2 \dots a_n$$

with length $n \in \mathbb{N}$:

1. Let $r_0, r_1, \dots, r_n$ be a sequence of states where $r_0 = q$ and $r_{i+1} = \delta(r_i, a_{i+1})$ for every integer i such that $0 \le i \le n-1$. Set $\delta^\star(q, \omega)$ to be the last state, r_n, in this sequence.

2. Apply the following ***recursive definition*** instead: For every state $q \in Q$ and for every string $\omega \in \Sigma^\star$,

$$\delta^\star(q, \omega) = \begin{cases} q & \text{if } \omega = \lambda, \\ \delta(\delta^\star(q, \mu), \tau) & \text{if } \omega = \mu \cdot \tau \text{ for a string } \mu \in \Sigma^\star \text{ and symbol } \tau \in \Sigma. \end{cases}$$

Supplemental notes for this lecture include a proof that these definitions are equivalent (that is, they describe *the same* state $\delta^\star(q, \omega)$ for every state $q \in Q$ and string $\omega \in \Sigma^\star$).

If You Want to Get Started. . .

If time permits, please consider applying the above deterministic finite automaton to several short strings in $\Sigma^\star$, including the empty string. Which strings are accepted? What are the states q_0, q_1 and q_2 being used to keep track of?

Can you see what the language of this deterministic finite automaton is? Is there a proof technique that, that you have already learned about, to ***proof*** that this deterministic finite automaton has the language that you guessed? Similarly, how can you ***prove*** that the two definitions of the "expected transition function", that have now been given, are equivalent?