<Lecture 2 ([illegible])>

Finite State Machine ADA

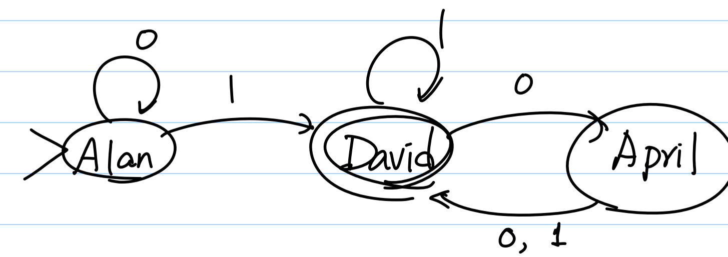
Upon receiving the pen, execute the following:

Alan : if input is empty, say "reject"
otherwise read and erase the first symbol
- 0 : do nothing
- 1 : give the pen to David

David : if input is empty, say "accept"
otherwise read and erase the first symbol
- 0 : give the pen to April
- 1 : do nothing

April : if input is empty, say "reject"
otherwise read and erase the first symbol
- 0, 1 : give the pen to David

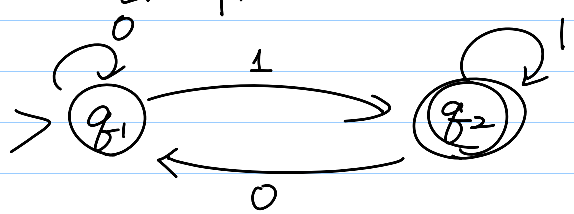
ex.

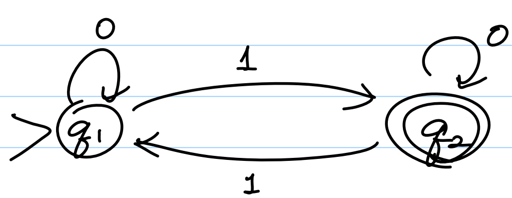


1101, 1, 01, 100 110001

0 10 1010

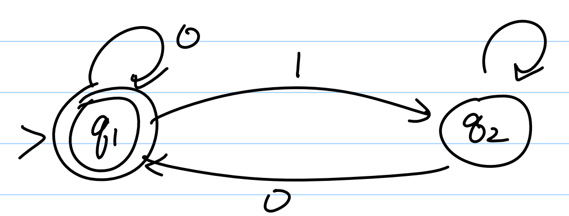
Another Example

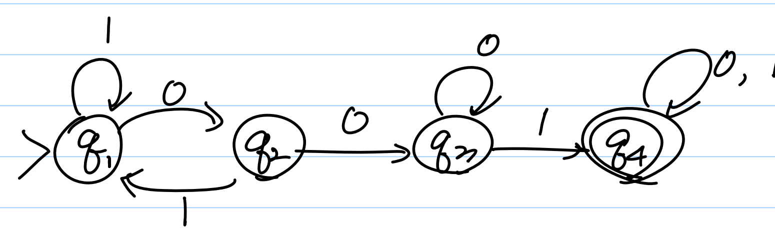




Q

Q.





Defining Finite Automata

1. informal definition (pictures)

2. formal definition (what we really need to understand!)

Before presenting the formal Definition,

Do you remember what are the formal definitions of functions. relations, cartesian product?

- Cartesian Product of sets A and B,

  $A \times B = \{(x, y) \mid x \in A,\ y \in B\}$

  $A = \{1, 2, 3\}$
  $B = \{a, b\}$

  $A \times B =$

- Relation: a subset of $A \times B$, i.e., a set of ordered pairs

  $A = \{1, 2, 3\}$
  $B = \{a, b\}$

  example?

- Function : a special case of a relation s.t. every element from A (called "domain") is associated with only one element from B (called "range").

$$A = \{1, 2, 3\}$$
$$B = \{a, b\}$$

example?

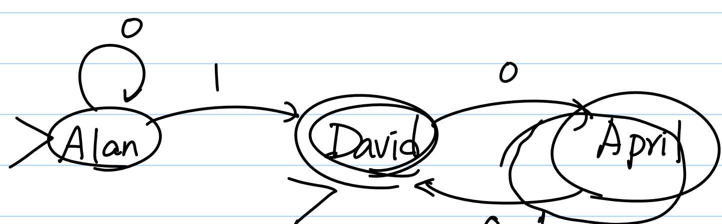
Often, we write it as $f : A \to B$

Formal Definition of DFA (finite automata)

A DFA is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ where

- Q is a finite set called the "states"
- Σ is a finite set called the "alphabet"
- $\delta : Q \times \Sigma \to Q$ is the "transition function"
- $q_0 \in Q$ is the start state
- $F \subseteq Q$ is the set of accept states

Example



$Q =$

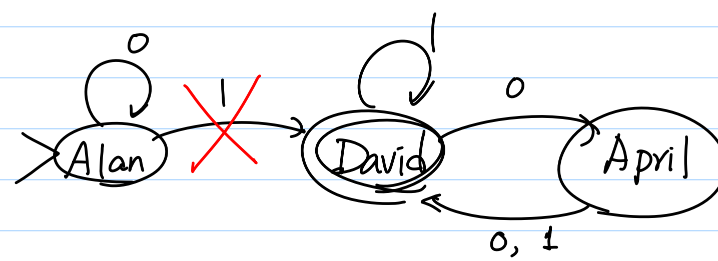
$\Sigma =$

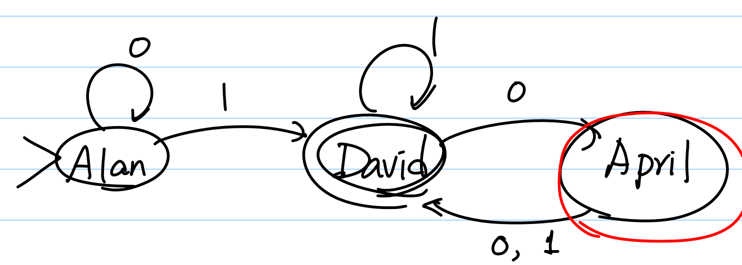
q_0 :

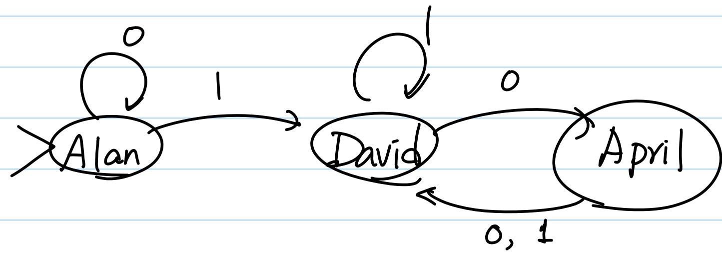
F

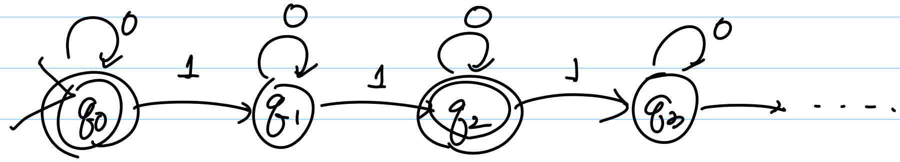
δ

Exercise









Formal Definition of "DFA M accepts input w" (p40)

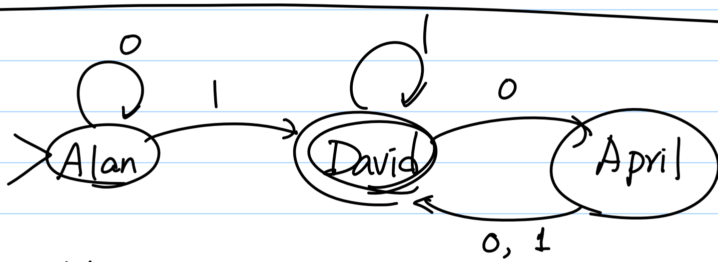
Intuition: The sequence starts with the correct start state and follow the "legal" transitions and ends in one of the accept states

Given $M = (Q, \Sigma, \delta, q_0, F)$ and $w = a_1 \cdots a_n$ $(a_i \in \Sigma)$, M accepts w if there is a sequence of states $\langle r_0, r_1, \cdots, r_n \rangle$ s.t.

1. $r_0 = q_0$
2. $\delta(r_i, a_{i+1}) = r_{i+1} \quad (i = 0, \cdots, n-1)$
3. $r_n \in F$

Given $M = (Q, \Sigma, \delta, q_0, F)$ and $w = a_1 \cdots a_n$ $(a_i \in \Sigma)$, M accepts w if there is a sequence of states $\langle r_0, r_1, \cdots, r_n \rangle$ s.t.

1. $r_0 = q_0$
2. $\delta(r_i, a_{i+1}) = r_{i+1} \quad (i = 0, \cdots, n-1)$
3. $r_n \in F$



1101

1010

Given $M = (Q, \Sigma, \delta, q_0, F)$ and $w = a_1 \cdots a_n$ $(a_i \in \Sigma)$, M accepts w if there is a sequence of states $\langle r_0, r_1, \cdots, r_n \rangle$ s.t.

1. $r_0 = q_0$
2. $\delta(r_i, a_{i+1}) = r_{i+1} \quad (i = 0, \cdots, n-1)$
3. $r_n \in F$

formal Def of ADA

1. $Q = \{$Alan, David, April$\}$
2. $\Sigma = \{0, 1\}$
3. δ

	0	1
Alan	alan	David
David	April	David
april	David	David

4. alan is the start state
5. David is the only accept state

1 1 0 1

1 0 1 0

Machine vs Language

- "M recognizes Language A" if
  $A = \{ w \mid M \text{ accepts } w \}$
- A is the language of M (written as $L(M) = A$) if A is the set of all strings that M accepts.
- A machine may accept several strings, but accept only one language
- If M does not accept any string, does it recognize no language?

A language is <u>regular</u> if there is a DFA that recognizes it.