CSE 105
THEORY OF COMPUTATION

Spring 2018

http://cseweb.ucsd.edu/classes/sp18/cse105-ab/
Today's learning goals

- Design an automaton that recognizes a given language.
- Specify each of the components in a formal definition of an automaton.
- Prove that an automaton recognizes a specific language.
Deterministic finite automaton

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

1. \(Q\) is a finite set called the states
2. \(\Sigma\) is a finite set called the alphabet
3. \(\delta: Q \times \Sigma \rightarrow Q\) is the transition function
4. \(q_0 \in Q\) is the start state
5. \(F \subseteq Q\) is the set of accept states.

Can there be more than one start state in a finite automaton?

A. Yes, because of line 4.
B. No, because of line 4.
C. I don't know
A finite automaton is a 5-tuple \((Q, \Sigma, \delta, q_0, F)\) where

1. \(Q\) is a finite set called the states
2. \(\Sigma\) is a finite set called the alphabet
3. \(\delta : Q \times \Sigma \rightarrow Q\) is the transition function
4. \(q_0 \in Q\) is the start state
5. \(F \subseteq Q\) is the set of accept states.

How many outgoing arrows from each state?

A. May be different number at each state.
B. Must be 2.
C. Must be \(|Q|\).
D. Must be \(|\Sigma|\).
E. I don't know.
An example

Define $M = (\{q_1, q_2, q_3, q_4\}, \{a, b\}, \delta, q_1, \{q_4\})$ where the function $\delta$ is specified by its table of values:

<table>
<thead>
<tr>
<th>Input in $Q \times \Sigma$</th>
<th>Output in $Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(q_1, a)$</td>
<td>$q_3$</td>
</tr>
<tr>
<td>$(q_2, a)$</td>
<td>$q_2$</td>
</tr>
<tr>
<td>$(q_3, a)$</td>
<td>$q_3$</td>
</tr>
<tr>
<td>$(q_4, a)$</td>
<td>$q_2$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input in $Q \times \Sigma$</th>
<th>Output in $Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(q_1, b)$</td>
<td>$q_2$</td>
</tr>
<tr>
<td>$(q_2, b)$</td>
<td>$q_2$</td>
</tr>
<tr>
<td>$(q_3, b)$</td>
<td>$q_4$</td>
</tr>
<tr>
<td>$(q_4, b)$</td>
<td>$q_4$</td>
</tr>
</tbody>
</table>

Draw the state diagram for the DFA with this formal definition.
An example

\( (\{q_1, q_2, q_3, q_4\}, \{a, b\}, \delta, q_1, \{q_4\}) \)

What's an example of a

- length 1 string accepted by this DFA?
- length 1 string rejected by this DFA?
- length 2 string accepted by this DFA?
- length 2 string rejected by this DFA?
An example

\(\{q1, q2, q3, q4\}, \{a, b\}, \delta, q1, \{q4\}\)

What's the best description of the language recognized by this DFA?

A. Starts with b and ends with a or b
B. Starts with a and ends with a or b
C. a's followed by b's
D. More than one of the above
E. I don't know.

and using set notation?
An example

\[ \{q_1, q_2, q_3, q_4\}, \{a, b\}, \delta, q_1, \{q_4\} \]

This DFA recognizes the language of all strings of the form a's followed by b's

i.e. \( \{ a^n b^k \mid n, k \geq 1 \} \)
An example \( \{ a^n b^k \mid n, k \geq 1 \} \)

Is this the same as the language described by
A. \( a^*b^* \)
B. \( a(ba)^*b \)
C. \( a^* U b^* \)
D. \( (aaa)^* \)
E. None of the above
Regular languages

- DFA $M$ over the alphabet $\Sigma$
  - For each string $w$ over $\Sigma$, $M$ either accepts $w$ or rejects $w$
  - The **language recognized by $M$** is the set of strings $M$ accepts, and is also known as the **language of $M$** and written $L(M)$

A language is **regular** iff there is some finite automaton that recognizes **exactly** it.
Building DFA

Definition: A language is **regular** means there is some DFA that recognizes it.

**Typical questions**
Define a DFA which recognizes the language L.

*or*

Prove that the (given) language L is regular.
Building DFA

Example
Define a DFA which recognizes

\{ w \mid w \text{ has at least 2 a's} \}

Bonus: what would you change if “at most” instead?
Justification?

To prove that the DFA we build, $M$, actually recognizes the language $L$

\[ \text{WTS } L(M) = L \]

(1) Is every string accepted by $M$ in $L$?

(2) Is every string from $L$ accepted by $M$?

*or contrapositive version:* Is every string rejected by $M$ not in $L$. 
Building DFA

Remember

States are our only (computer) memory.

Design and pick states with specific roles / tasks in mind.

"Have not seen any of desired pattern yet"
"Trap state"
Specifying an automaton

( \{q1,q2,q3\}, \{a,b\}, \delta, q1, ? )

What state(s) should be in F so that the language of this machine is \{ w | ab is a substring of w}\?

A. \{q2\}  
B. \{q3\}  
C. \{q1,q2\}  
D. \{q1,q3\}  
E. I don't know.
Specifying an automaton

( \{q1,q2,q3\}, \{a,b\}, \delta, q1, ? )

What state(s) should be in F so that the language of this machine is \{ w \mid b's never occur after a's in w\}?

A. \{q2\}
B. \{q3\}
C. \{q1,q2\}
D. \{q1,q3\}
E. I don't know.
Building DFA

Formally,

\{w \mid w \text{ contains the substring } ab\}

\{w \mid w \text{ doesn't contain the substring } ab\}
For next time

- Finish Individual Homework 0 due Saturday
- Review quiz 1 due Sunday (for credit)
- Read Individual Homework 1 due Tuesday

Pre class-reading for Monday:
Theorem 1.25, Theorem 1.26
Complementation

Claim: If $A$ is a regular language over $\{0,1\}^*$, then so is $\overline{A}$
aka "the class of regular languages is closed under complementation"