Today's learning goals

- Use and design a finite automaton via its
  - Formal definition
  - State diagram
- Identify the strings and languages accepted by a given finite automaton
- Design a finite automaton which accepts a given language
- Define the regular operations on languages
- Prove closure properties of the class of regular languages
Review

• **Alphabet**: nonempty finite set of **symbols**
• **String over an alphabet**: finite sequence of symbols
• **Language over an alphabet**: some set of strings

• **DFA over an alphabet**: deterministic finite automaton
  • Input: finite string over a fixed alphabet
  • Output: "accept" or "reject"
  • $L(M) = \{w \mid M \text{ accepts } w\}$
• **Regular language** language that is $L(M)$ for some DFA $M$
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. \(\delta : Q \times \Sigma \to Q\) 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.
Regular languages

- If $A$ is the set of strings that DFA $M$ recognizes (accepts)
  - We say $A$ is the language of $M$
  - We write $L(M) = A$
  - We have that $A$ is regular because…

A language is regular if there is some finite automaton that recognizes exactly it.
An example

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 \}
Another example

What is the best description of language recognized by this automaton?

A. \{ a^n b^k \mid n,k \geq 1 \}
B. \{ a^n b^k \mid n \geq 1, b \geq 0 \}
C. \{ awb \mid w \text{ in } \{a,b\}^* \}
D. \{ aw \mid w \text{ in } \{a,b\}^* \}
E. I don't know
Specifying an automaton

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

What's the best representation of \( \delta \) for this DFA?
A. \( q1 \rightarrow b, q1 \rightarrow a, q2 \rightarrow a, q2 \rightarrow b, q3 \rightarrow a,b. \)
B. \( \{ (q1,b,q1),(q1,a,q2),(q2,a,q2),(q2,b,q3), (q3,a,q3),(q3,b,q3) \} \)
C. \( \delta(b) = \text{same}, \delta(a) = \text{change} \)
D. No description other than the state diagram (circles & arrows) is possible.

A. 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 | \text{ab is a substring of w}\}?  

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

\text{\textit{Note: we} \{a, b\}^*}
Specifying an automaton

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

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

A. \{ q_2 \}
B. \{ q_3 \}
C. \{ q_1, q_2 \}
D. \{ q_1, q_3 \}
E. I don't know.
Recall terminology

- **Alphabet**: nonempty finite set of symbols
- **String over an alphabet**: finite sequence of symbols
- **Language over an alphabet**: some set of strings

- **DFA over an alphabet**: deterministic finite automaton
  - Input: finite string over a fixed alphabet
  - Output: "accept" or "reject"
  - \( L(M) = \{ w \mid M \text{ accepts } w \} \)
- **Regular language**
  - language that is \( L(M) \) for some DFA \( M \)
A useful (optional) bit of terminology

When is a string accepted by a DFA?

$\delta^*(q_0, w) = \delta(\delta(q_0, w_0), w_1), \ldots$

**Computation of $M$ on $w$:** where do we land when start at $q_0$ and read each symbol of $w$ one-at-a-time?

$\bar{\delta}^*(q, w) =$

- state
- string

$\begin{cases} q \\ \delta \left( \delta^*(q, u), x \right) \end{cases}$

if

- $w = \underbrace{u \ldots u}_n$ where $\Sigma \ni x \in \Sigma$
- $w = \underbrace{u \ldots u}_n$ for some $\Sigma \ni x \in \Sigma$

String
Is there an infinite regular language?

A. No: all regular languages have to be finite.
B. Yes: all regular sets are infinite.
C. Yes: all infinite sets of strings over an alphabet are regular.
D. Yes: some infinite sets of strings over each alphabet are regular and some are not.
E. I don't know.
Is every finite language regular?

A. No: some finite languages are regular, and some are not.
B. No: there are no finite regular languages.
C. Yes: every finite language is regular.
D. I don't know.
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"
For next time

Homework 1 due tomorrow

- Set up course tools: Gradescope, Piazza, JFLAP, ieng6
- Keep working
- Ask questions in office hours