Indiv Th1 due tomorrow
note: E "as a binary number"
represents 0
Today's learning goals

- Prove closure properties of the class of regular languages
- Apply closure properties to conclude that a language is or isn't regular

Regular Langs

$L_2 = L(R)$

$OP \subseteq L_2$

$OP \subseteq L(M)$

$L \in L(M)$
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
    a.k.a. the language of $M$ is the set of strings $M$ accepts
    a.k.a. $L(M) = \{ w \mid w$ is a string over $\Sigma$ and $M$ accepts $w\}$

A language is regular iff there is some finite automaton that recognizes exactly it.
Justification?

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

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 $L$?
A useful (optional) bit of terminology

When is a string accepted by a DFA?

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

$M = (Q, \Sigma, \delta, q_0, F)$

$\delta^*( (q, w) ) =$

Recursively defined function
Building DFA

Formally,

\[ A = \{ w \mid w \text{ contains the substring } ab \} \]

\[ \overline{A} = \{ w \mid w \text{ doesn't contain the substring } ab \} \]

\[ = \{ w \mid \text{all b's in w come before a's} \} \]

\[ = L(b^* a^*) = L(b^* b^* a^* a^*) \]
Complementation

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

Proof: Let $A$ be a regular language. Then there is a DFA $M = (Q, \Sigma, \delta, q_0, F)$ such that $L(M) = A$. We want to build a DFA whose language is $\overline{A}$. Define $\overline{M'} = (Q, \Sigma, \delta', q_0, F')$

Claim of Correctness $L(M') = \overline{A}$

Proof of claim…
1. For all strings \( w \), if \( w \in L(M') \)
   then \( w \in \overline{A} \)

Let \( w \) be arbitrary string, assume \( M' \) accepts \( w \).
By def \( \delta^* ((q_0, w)) \in Q - F \)
But \( \delta^* ((q_0, w)) \) is the last state reached of \( M \) on \( w \).
So since this state is not in \( F \), which is set of accept states of \( M \), \( M \) rejects \( w \).
By def \( L(M) = \overline{A} \) so \( w \notin \overline{A} \)
\( \therefore w \notin A \)
Why closure proofs?

- General technique of proving a new language is regular
- Stretch the power of the model
- Puzzle!
Set operations

Input language (s) $\rightarrow$ OPERATION $\rightarrow$ Output language

The class of regular languages is closed under …

- Complementation ✔
- Kleene star ?
- Concatenation ?
- Union ?
- Intersection ?
- Set difference ?
Union

Theorem: The class of regular languages over fixed alphabet $\Sigma$ is closed under the union operation.

Proof: What are we proving here?

A. For any set $A$, if $A$ is regular then so is $A \cup A$.
B. For any sets $A$ and $B$, if $A \cup B$ is regular, then so is $A$.
C. For two DFAs $M_1$ and $M_2$, $M_1 \cup M_2$ is regular.
D. None of the above.
E. I don’t know.
Theorem: The class of regular languages over fixed alphabet $\Sigma$ is closed under the union operation.

Proof: Let $A_1, A_2$ be any two regular languages over $\Sigma$. WTS that $A_1 \cup A_2$ is regular.

Goal: build a machine that recognizes $A_1 \cup A_2$. 
Union

Sipser Theorem 1.25 p. 45

Goal: build a machine that recognizes $A_1 \cup A_2$.

Strategy: use machines that recognize each of $A_1$, $A_2$.

**HOW?**
At start of computation

$w = @bbaba$

The diagram shows a transition diagram for a machine with states $q_0, q_1, q_2, q_3$, and $q_f$. The initial state is $q_0$, and the machine transitions to $q_1$ on input $b$, to $q_2$ on input $a$, and to $q_3$ on input $b$. The machine's state changes based on the input sequence $w$. The top machine's state transitions are marked with red arrows, indicating the machine's state changes based on the input sequence. The bottom machine's state transitions are marked with blue arrows, showing the state changes not covered by the red arrows. The transition functions are labeled appropriately, with $q_i, q_f$ indicating final states.
"Run in parallel"

Building $M = (Q_1 \times Q_2, \Sigma, \delta, ?, ?)$

Given $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$  $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$

Start state:
Accept state(s):
Transition function:
"Run in parallel"

Building $M = (Q_1 \times Q_2, \Sigma, \delta, ?, ?)$

Given $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$

Start state:
Accept state(s):
Transition function:

The set of accepting states for $M$ is

A. $F_1 \times F_2$
B. $\{ (r,s) \mid r \text{ is in } F_1 \text{ and } s \text{ is in } F_2 \}$
C. $\{ (r,s) \mid r \text{ is in } F_1 \text{ or } s \text{ is in } F_2 \}$
D. $F_1 \cup F_2$
E. I don't know.
"Run in parallel"

Building $M = (Q_1 \times Q_2, \Sigma, \delta, ?, ?)$

Given $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$

Start state:

Accept state(s):

Transition function:
Proof: Let $A_1, A_2$ be any two regular languages over $\Sigma$. Given $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ such that $L(M_1) = A_1$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ such that $L(M_2) = A_2$.

WTS that $A_1 \cup A_2$ is regular.

Define $M = (Q_1 \times Q_2, \Sigma, \delta, (q_1, q_2), \{(r, s) \in Q_1 \times Q_2 \mid r \in F_1 \text{ or } s \in F_2\})$ with $\delta((r, s), x) = (\delta_1(r, x), \delta_2(s, x))$ for $(r, s) \in Q_1 \times Q_2$, $x \in \Sigma$.

Why does $L(M) = A_1 \cup A_2$?
Aside: Intersection

• How would you prove that the class of regular languages is closed under intersection?
• Can you think of more than one proof strategy?

\[ A \cap B = \{ x \mid x \text{ in } A \text{ and } x \text{ in } B \} \]
General proof structure/strategy

Theorem: For any $L$ over $\Sigma$, if $L$ is regular then [the result of some operation on $L$] is also regular.

Proof:

Given name variables for sets, machines assumed to exist.

WTS state goal and outline plan.

Construction using objects previously defined + new tools working towards goal. Give formal definition and explain.

Correctness prove that construction works.

Conclusion recap what you've proved.
Before/For next time

- Work on Individual Homework 1 due Tuesday
- Work on Group Homework 1 due Saturday

Pre class-reading for Wednesday:
  - Page 48 (Figure 1.27 and description below it)
  - Example 1.35 on page 52
Payoff

\{ w \mid \text{w contains neither the substrings aba nor baab} \}

Is this a regular set?
Payoff

\{ w \mid w \text{ contains neither the substrings } aba \text{ nor } baab \}\]

Is this a regular set?

A = \{ w \mid w \text{ contains } aba \text{ as a substring} \}
B = \{ w \mid w \text{ contains } baab \text{ as a substring} \}

\overline{A \cap B} = \overline{A} \cup \overline{B}