CSE 105
THEORY OF COMPUTATION

Spring 2018

http://cseweb.ucsd.edu/classes/sp18/cse105-ab/

Reminders:
Indiv HW1 due tomorrow.

Note: \( \varepsilon \) represents the value 0 as a binary number.
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 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. $L(M) = \{ w \mid w \text{ is a string over } \Sigma \text{ and } M \text{ accepts } w \}$

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

$L = L(M)$
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?
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?*

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

$\delta^*( (q, w) ) = \begin{cases} \emptyset & \text{if } w = \varepsilon \\ \delta((\delta^*(q, x), a)) & \text{if } w = xa \\ \delta((\delta^*(q, x), a)) & \text{if } w \in \Sigma^* \text{ and } a \in \Sigma \end{cases}$

Source state

$w \in \Sigma^*$
Formally, 

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

\[ = L((a \cup b)^*a_b (a \cup b)^*) \]

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

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

\[ ab \notin A \]

\[ ab \notin L(M_2) \]
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"
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

$$M' = \text{[DFA diagram]}$$

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

**Proof of claim…**
For each $w$, if $w \in L(M')$ then $w \in \overline{A}$.

Let $w$ be a string, assume $w \in L(M') \subseteq \{ \text{strings accepted by } M' \}$.

Since $M'$ accepts $w$, $\delta^*( (q_0, w) ) \in Q - F$.

How does $M$ behave on $w$? (b/c $L(M) = A$)

By def. computation of $M$ ends on

$\delta^* ( (q_0, w) )$ (just like $M'$) so $M$ rejects $w$.

So since $L(M) = A$, $w \not\in A$, i.e., $w \in \overline{A}$.

WTS 2 For each $w$, if $w \in L(M')$ then $w \notin \overline{A}$.
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 ?
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.
Union

Sipser Theorem 1.25 p. 45

**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 $\Sigma$. \textbf{WTS} that $A_1 \cup A_2$ is regular.

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

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

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

**HOW?**

Input $\xrightarrow{}$ M1, M2

Accept if either (or both) accepts

**HOW?**
\[
\mathcal{L}(A) = \{ w \mid \text{w don't have } ab \text{ as substring} \}
\]

\[
\mathcal{L}(B) = \{ w \mid \text{w is even} \}
\]

\[A \cup B = \{ w \mid \text{w has even length} \}
\]

\[aabaab \notin \]
"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?

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

**Theorem:** For any L over Σ, 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 w \text{ contains neither the substrings } \text{aba} \text{ nor } \text{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 \overline{B} = \overline{A \cup B}