Today's learning goals

- Review what it means for a set to be closed under an operation.
- Define the regular operations on languages.
- Prove closure properties of the class of regular languages.

Sipser Ch 1.1, 1.2
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"
Building DFA

New strategy

Express $L$ in terms of simpler languages – use them as building blocks.

Example

$L = \{ w \mid w \text{ does not contain the substring baba} \}$

$= \text{the complement of the set}$

$\{w \mid w \text{ contains the substring baba}\}$
Building DFA

DFA recognizing \{w \mid \text{w contains the substring baba}\}

DFA recognizing \{w \mid \text{w doesn't contain the substring baba}\}

modify (1)

interchange acc + rej states
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"
Closure of ... under ...

- \( \mathbb{Z} \) under addition.
- Set of even ints under multiplication.
- \( \{0\}^* \) under concatenation.

Which of these is true?

A. The set of odd integers is closed under addition.
B. The set of positive integers is closed under subtraction.
C. The set of rational numbers is closed under multiplication.
D. The set of real numbers is closed under division.
E. I don't know.
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"
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"

**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 $M' = \ldots$

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

**Proof of claim**...
\[
\text{Goal: } L(M^') = \overline{A}
\]

(1) \(\overline{A} \subseteq L(M^')\)

Let \(s \in \overline{A}\). Then \(s \notin A\). Then the computation of \(M\) on input \(s\) will end in a non-accepting state because \(s \notin A\) and \(A = L(M) = \text{set of strings that } M\text{ accepts}\). Then, the computation of \(M^'\) on input \(s\) will end in an accepting state because the computation doesn't change from \(M\) to \(M^'\), only the set of accepting states. Since accept states of \(M^'\) are the non-accept states of \(M\), \(M^'\) accepts \(s\), so \(s \in L(M^')\).

(2) \(L(M^') \subseteq \overline{A}\)

If \(s \in L(M^')\), then \(s \notin \overline{A}\).

Contrapositive:

If \(s \notin \overline{A}\), then \(s \notin L(M^')\).
Why closure proofs?

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

For $A, B$ languages over same alphabet, define:

$A \cup B = \{ x | x \in A \text{ or } x \in B \}$

$A \circ B = \{ xy \mid x \in A \text{ and } y \in B \}$

$A^* = \{ x_1 x_2 \ldots x_k \mid k \geq 0 \text{ and each } x_i \in A \}$

These are operations on sets!
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.
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? **
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 $\Sigma$. 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 $L(M_2) = A_2$ and WTS that $A_1 \cup A_2$ is regular.

Define $M = (?, \Sigma, \delta, ?, ?)$
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 $\Sigma$. 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 $L(M_2) = A_2$ and WTS that $A_1 \cup A_2$ is regular.

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

Idea: run in parallel
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 $\Sigma$. 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 $L(M_2) = A_2$ and WTS that $A_1 \cup A_2$ is regular.

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

What should be the initial state of $M$?

A. $q_0$
B. $q_1$
C. $q_2$
D. $(q_1, q_2)$
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 $\Sigma$. 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 $L(M_2) = A_2$ and WTS that $A_1 \cup A_2$ is regular. Define $M = (Q_1 \times Q_2, \Sigma, \delta, ?, ?)$

When $r$ is a state in $M_1$, $s$ is a state in $M_2$, and $x$ is in $\Sigma$, then $\delta((r,s), x) =$

A. $(r,s)$
B. $\left( \delta_1(r,x), \delta_2(s,x) \right)$
C. $\left( \delta_1(r,x), s \right)$
D. $\left( \delta_1(r,x), \delta_2(s,x) \right)$
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 $\Sigma$.

Given $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ such that $L(M_1) = A_1$ and $L(M_2) = A_2$.

WTS that $A_1 \cup A_2$ is regular.

Define $M = (Q_1 \times Q_2, \Sigma, \delta, (q_1, q_2), F_1 \times F_2)$.

The set of accepting states for $M$ is $F_1 \times F_2$.

A. $F_1 \times F_2$

B. $\{ (r, s) | r \text{ is in } F_1 \text{ and } s \text{ is in } F_2 \}$

C. $\{ (r, s) | r \text{ is in } F_1 \text{ or } s \text{ is in } F_2 \}$

D. $F_1 \cup F_2$

E. I don't know.
Union

Sipser Theorem 1.25 p. 45

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 each $(r, s) \in Q_1 \times Q_2$ and $x \in \Sigma$.

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

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

\[ A \cap B = \{ x | x \text{ in } A \text{ and } x \text{ in } B \} \]
Payoff

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

Is this a regular set?
Payoff

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

Is this a regular set?

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

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

\overline{A} \cap \overline{B} = \overline{A \cup B}
Sample closure proofs

• The class of regular languages over \(\{0,1\}\) is closed under the FlipBits operation, where

\[
\text{FlipBits}(L) = \{ w \mid w \text{ is obtained from some } w' \text{ in } L \text{ by flipping each } 0 \text{ in } w \text{ to } 1, \text{ and each } 1 \text{ to } 0 \}
\]

Ex.: \(L = \{011, 101\} \Rightarrow \text{FlipBits}(L) = \{100, 010\}\)

• The class of regular languages of \(\{a,b,z\}\) is closed under the DeleteWordsWithZ operation, where

\[
\text{DeleteWordsWithZ}(L) = \{ w \mid w \text{ is in } L \text{ and } w \text{ doesn't contain } z \}
\]
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.
The regular operations

For $A$, $B$ languages over same alphabet, define:

$$A \cup B = \{x \mid x \in A \text{ or } x \in B\}$$

$$A \circ B = \{xy \mid x \in A \text{ and } y \in B\}$$

$$A^* = \{x_1x_2\ldots x_k \mid k \geq 0 \text{ and each } x_i \in A\}$$

How can we prove that the concatenation of two regular languages is a regular language?