CSE 105
THEORY OF COMPUTATION

"Winter" 2018

http://cseweb.ucsd.edu/classes/wi18/cse105-ab/

Today's learning goals

Sipser Section 1.1

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

Regular languages

Sipser p. 35 Def 1.5

- DFA M over the alphabet Σ
  - For each string w over Σ, 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 | w is a string over Σ 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?*

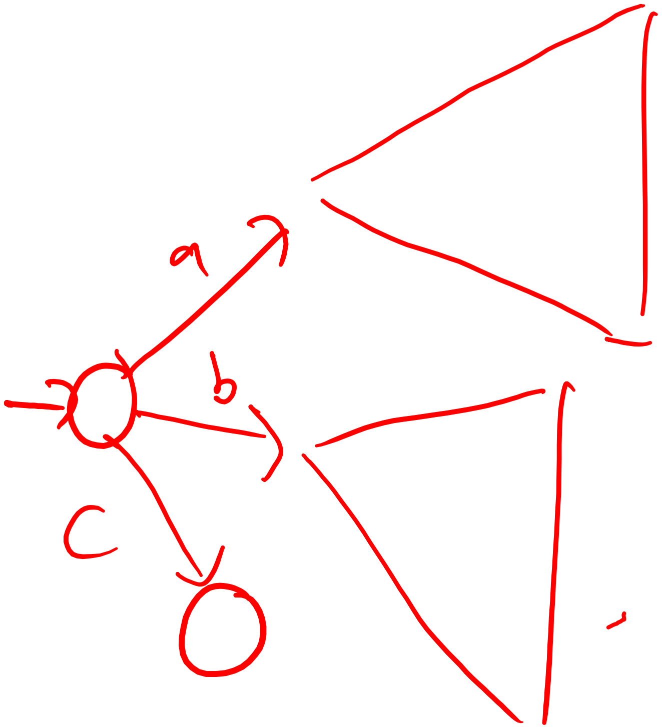
$$\delta^*(q, w) = \begin{cases} q & \text{if } w = \varepsilon \\ \text{let } w = av \text{ where } a \text{ is a symbol} \\ \delta^*(\delta(q,a), v) & \end{cases}$$

Recursively defined function

Regular languages: bounds?

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.


a
b
c

Building DFA

Remember

States are our only (computer) memory.

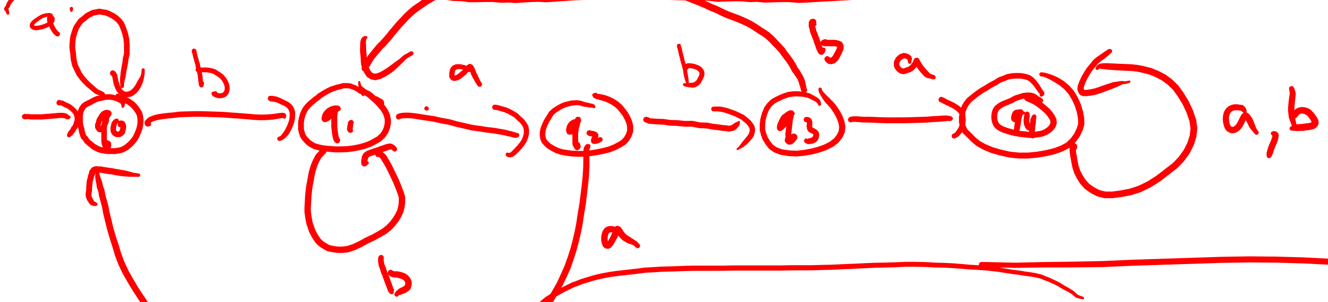
Design and pick states with specific roles / tasks in mind.

"Have not seen any of desired pattern yet"

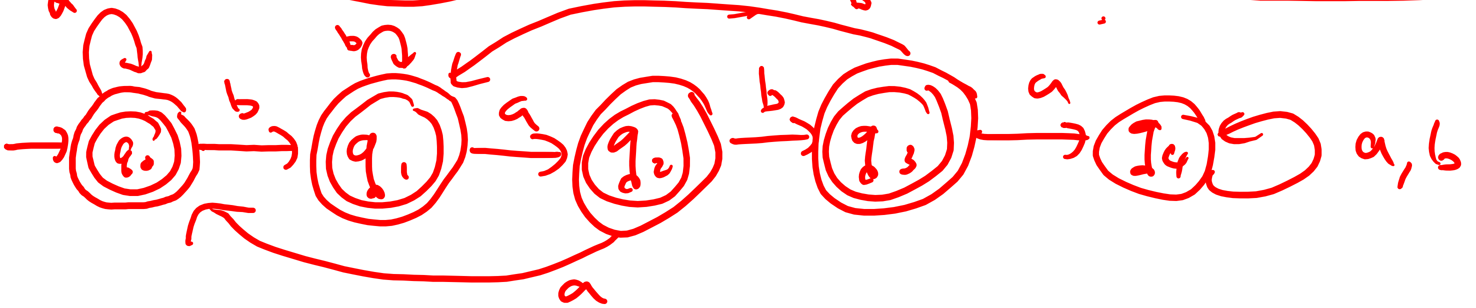
"Trap state"

Building DFA

DFA recognizing {w | w contains the substring baba}



DFA recognizing {w | w doesn't contain the substring baba}



Building DFA

New strategy

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

Example

L = { w | w does not contain the substring baba }

= the complement of the set

{w | w contains the substring baba}

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"

<u>proof idea</u>: Let A be an arbitrary reg language.

let M be a DFA that recognizes A.

$\boxed{\text{Construct } M' \text{ that recognizes } \overline{A}.}$

$\Rightarrow \overline{A}$ is regular

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 $M=(Q,\Sigma,\delta,q0,F)$ such that $L(M) = A$. We want to build a DFA whose language is $\overline{A}$. Define

M' = $\boxed{?}$

$(\underline{Q}\ \underline{\Sigma}\ \underline{\delta}\ \underline{q0}\ \underline{F})$

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

Proof of claim…

$L(M') \subseteq \overline{A}$

$\overline{A} \subseteq L(M')$

$L(M') \subseteq \overline{A}$. Suppose $w \in L(M')$

$\Rightarrow \delta^*(q_0, w) = f$ for some $f \in \overline{F}$

$f \notin F$.

$\Rightarrow w \notin L(M)$, $\Rightarrow w \notin A \Rightarrow w \in \overline{A}$

$\overline{A} \subseteq L(M')$. Suppose $w \in \overline{A}$. $\Rightarrow w \notin A$

$\Rightarrow \delta^*(q_0, w) = g$ for $g \notin F$

_ _ _ _ _ _ _ Finish.

Why closure proofs?

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

Set operations

Input set(s) → OPERATION → Output set

- Complementation ✓
- Kleene star
- Concatenation
- Union
- Intersection
- Set difference

The regular operations

Sipser Def 1.23 p. 44

For A, B languages over same alphabet, define:

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

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

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

These are operations on sets!

Union

Sipser Theorem 1.25 p. 45

Theorem: The class of regular languages over fixed alphabet Σ is closed under the **union operation**.

for any reg. languages A, B, A ∪ B is regular.

Proof:

What are we proving here?

A. For any set A, if A is regular then so is A U A.
B. For any sets A and B, if A U B is regular, then so is A.
C. ~~For two DFAs M1 and M2, M1 U M2 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 Σ is closed under the union operation.

Proof: Let A1, A2 be any two regular languages over Σ.

WTS that A1 U A2 is regular.

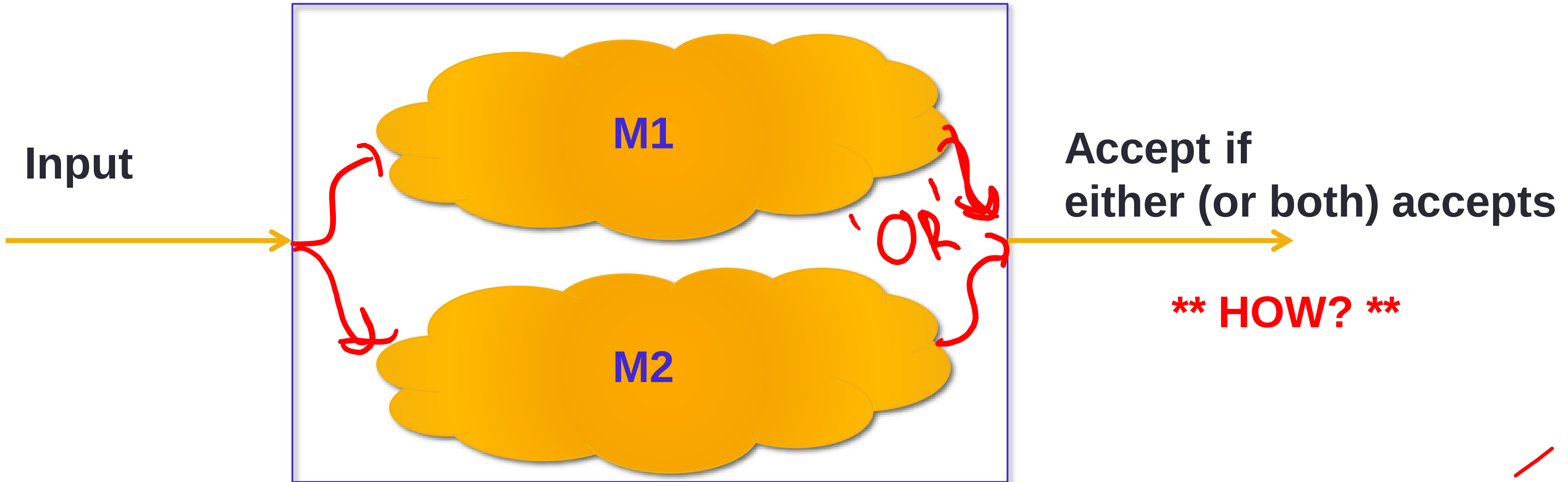
Goal: build a machine that recognizes A1 U A2.

Union

Sipser Theorem 1.25 p. 45

Goal: **build a machine that recognizes A1 U A2.**

Strategy: **use machines that recognize each of A1, A2.**



"Run in parallel"

$\delta((q,p),a) = (\delta_1(q,a), \delta_2(p,a))$

M = (Q1xQ2, Σ, δ, ?, ?)

Start state: (q_1, q_2) where q_1 is start of M_1, q_2 is start of M_2

Accept state(s):

Transition function:

The set of accepting states for M is

A. F1 x F2
B. { (r,s) | r is in F1 and s is in F2 }
C. { (r,s) | r is in F1 or s is in F2 }
D. F1 U F2
E. I don't know.

Union

Sipser Theorem 1.25 p. 45

Proof: Let A1, A2 be any two regular languages over Σ.
Given M1 = (Q1,Σ,δ1,q1,F1) such that L(M1) = A1 and
M2 = (Q2,Σ,δ2,q2,F2) such that L(M2) = A2.

WTS that A1 U A2 is regular.

Define
M = (Q1xQ2,Σ,δ,(q1,q2),{(r,s) in Q1xQ2 | r in F1 or s in F2})
with δ((r,s), x) = (δ1(r,x), δ2(s,x)) for (r,s) in Q1xQ2 and x in Σ.
Why does L(M) = A1 U A2 ?

let $w \in A_1 \cup A_2$. then $w \in A_1$ or $w \in A_2$.

Case 1: $w \in A_1 \Rightarrow w \in L(M_1) \Rightarrow \delta_1^*(q_1, w) = f \in F_1$

$$\delta^*((q_1, q_2), w) = (\delta_1^*(q_1, w), \delta_2^*(q_2, w))$$

$$= (f, ?) \in F$$

$$= \{(r,s) \mid r \in F_1 \text{ or } s \in F_2\}$$

other direction exercise

(Hint: show if $w \notin A_1 \cup A_2$ then $w \notin L(M)$)

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?*

$$A \cap B = \{ x \mid x \text{ in } A \text{ and } x \text{ in } B \}$$

DeMorgan's Law

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

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

Payoff

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

Is this a regular set?

Payoff

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

Is this a regular set?

A = { w | w contains aba as a substring}

B = { w | w contains baab as a substring}

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

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.

The regular operations

Sipser Def 1.23 p. 44

For A, B languages over same alphabet, define:

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

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

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

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

For next time

- Work on Group Homework 1 **due Saturday**

Pre class-reading for Friday:

- Page 48 (Figure 1.27 and description below it)
- Example 1.35 on page 52