CSE 105
THEORY OF COMPUTATION

"Winter" 2018

http://cseweb.ucsd.edu/classes/wi18/cse105-ab/
Today's learning goals

- Design an automaton that recognizes a given language.
- Specify each of the components in a formal definition of an automaton.
- Prove that an automaton recognizes a specific language.
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 \(\{ w \mid \text{ab is a substring of } w\}\)?

A. \(\{q_2\}\)
B. \(\{q_3\}\)
C. \(\{q_1,q_2\}\)
D. \(\{q_1,q_3\}\)
E. I don't know.
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 \mid 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.
Building DFA

Typical questions
Define a DFA which recognizes the given language $L$.

or

Prove that the (given) language $L$ is regular.
Building DFA

Example

Define a DFA which recognizes

\{ w \mid w \text{ has at least 2 a's} \}
Building DFA

Example

Define a DFA which recognizes

\{ w \mid w \text{ has at most 2 a's} \}
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"
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?

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

Recursively defined function
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
  - The **language of** $M$ is the set of strings $M$ accepts
  - $L(M) = \{ w \mid w$ is a string over $\Sigma$ and $M$ accepts $w\}$

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

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

= the complement of the set

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

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

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

$$M' = \text{?}$$

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

Proof of claim…
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

For A, B languages over same alphabet, define:

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

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

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

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.
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 $A_1 \cup A_2$. 
For next time

• Finish Individual Homework 0 due Saturday
• Review quiz 1 due Sunday (for credit)
• Read Individual Homework 1 due Tuesday

Pre class-reading for Wednesday:
Theorem 1.25, Theorem 1.26