Mary gave a cat to John

Professor Jeanne Ferrante
Today’s agenda

• Formal definition of DFA
• DFA design
• Regular languages
• Closure properties of the regular languages

Lots More Office Hours! Check out calendar & Go!
Reminder:
• HW 1 due Friday (April 1, no fooling) by 11:59 pm
Lec. 1 Review Question

Which ONE of the following is FALSE?

A. There is a DFA whose language is the empty set
B. DFA’s have only 1 start state
C. DFA’s can recognize languages with infinitely many strings
D. DFA’s have only 1 final state
E. For any finite set of strings, there is a DFA that recognizes that finite set
A different way of specifying a DFA

FORMAL DEFINITION OF DFA
Formal Definition of a DFA

A DFA $M_1$ is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where:

- $Q$ is a finite set of states
- $\Sigma$ is a finite set of characters, the alphabet
- $\delta: Q \times \Sigma \rightarrow Q$, the transition function
- $q_0$, a member of $Q$, the start state
- $F$, a subset of $Q$, the set of final state(s)
Can 2 different DFA’s recognize the same language?

A. Yes, but the diagrams must be isomorphic as graphs
B. Yes, but the 2 automata must have the same number of states
C. Yes, and the automata can have a different number of states
D. No, each DFA recognizes a distinct language
DFA’s recognize Regular Languages

A language $L$ is **regular** if there is DFA that recognizes it.

- We’ll use the *formal definition* of DFA in proofs instead of diagrams
  - We won’t have DFA diagrams to work with
  - But we’ll be able to refer to specific parts of the DFA (e.g. $F$, the set of final states)

- Let’s take a closer look at $\delta$...
\[ \delta : Q \times \Sigma \rightarrow Q \]

Transition Table

\[ M = \]

\[ L(M) = \]
\{ w \mid \text{length}(w) \text{ is odd} \}
(Note: |w| = length of w)

Which table best describes \( \delta \)?

A.

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>q1</td>
<td>q1</td>
<td>q2</td>
</tr>
<tr>
<td>q2</td>
<td>q1</td>
<td>q2</td>
</tr>
</tbody>
</table>

B.

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>q1</td>
<td>q2</td>
<td>q2</td>
</tr>
<tr>
<td>q2</td>
<td>q1</td>
<td>q1</td>
</tr>
</tbody>
</table>

C.

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>q1</td>
<td>q1</td>
<td>q1</td>
</tr>
<tr>
<td>q2</td>
<td>q2</td>
<td>q2</td>
</tr>
</tbody>
</table>
Finite Automata: Historical Perspective

- In 1976, Rabin and Scott won the Turing Award, the highest award in computer science, for their 1959 paper, "Finite Automata and Their Decision Problems"
  - *They wrote the paper while spending a summer at IBM Research!*

- Origin of FA from 1943 McCulloch and Pitts paper, ""A logical calculus of the ideas immanent in nervous activity", a mathematical model of neural activity and networks
Now that you see how DFA’s work

DESIGNING YOUR OWN DFA
Which states should be in $F$ so that DFA $M_2$ recognizes the language $L_2 = \{w \mid w \text{ contains less than 1 } a \text{ or more than 1 } a\}$? ($\Sigma = \{a, b\}$)

$M_2$

A. $F = \{q2\}$
B. $F = \{q3\}$
C. $F = \{q1, q2\}$
D. $F = \{q1, q3\}$
E. $F = \{q2, q3\}$
Designing DFA’s

• States are the only mechanism for a DFA to “remember” what it has seen of input string so far

• When you design a DFA, it helps to associate a “meaning” to each state and name it accordingly
  – It’s comparable to writing comments in code

• It’s also useful, and we will practice, describing DFA transitions in words

• To be correct, need to satisfy both:
  – Include all strings you need to accept
  – Exclude all strings you need to reject (not accept)
Designing DFA’s

L = \{w \mid w \text{ contains aa as substring}\}

\Sigma = \{a, b\}

Please note that this diagram is correct, but would not execute correctly in JFLAP!
Designing DFA’s

$L = \{ w \mid w \text{ contains } aa \text{ as substring but not } bb \}$
Designing DFA’s

$L = \{ w \mid w \text{ is positive and even when interpreted as binary number} \}$

$\Sigma = \{0,1\}$
Digging deeper into what it means to be a regular language

CLOSURE PROPERTIES OF THE REGULAR LANGUAGES
Regular Languages

• The set of languages for which there exists a DFA that (exactly) recognizes the language
• Note that the Regular Languages is a set of sets of strings

• How large is this set?
• What are its boundaries?
• Thm 1.25. The class of regular languages is closed under the union operation.
Review: Closure Properties/”Closed Under”

• Which of the following statements are FALSE?

A. Integers are closed under addition
B. Integers are closed under division
C. Even numbers are closed under addition
Thm 1.25. The class of regular languages is closed under the union operation.

- \{ w \mid \text{in w, b's never appear after a's} \}

- \{ w \mid \text{length(w) is odd} \}

DESIGN METHOD:
Use Cartesian Product of State Sets
Thm. The class of regular languages is closed under the union operation.

Our working example:

A generalized proof form:

- Given two regular languages $L_1, L_2$, with DFAs $M_1, M_2$
  - $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$
  - $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$

- We want to construct a DFA $M = (Q, \Sigma, \delta, q_0, F)$, s.t.:
  - $Q = Q_1 \times Q_2$
  - $\delta = \delta_1 \cup \delta_2$
  - $q_0 = q_{01} \times q_{02}$
  - $F = F_1 \cup F_2$

- $M$ recognizes $L_1 \cup L_2$:
- A DFA recognizes $L_1 \cup L_2$, then $L_1 \cup L_2$ must be regular
Thm. The class of regular languages is closed under the union operation.

Proof:

Given: Two regular languages L1, L2, with DFAs M1, M2
   – M1 = (Q1, Σ, δ1, s1, F1)
   – M2 = (Q2, Σ, δ2, s2, F2)

We construct a DFA M = (Q, Σ, δ, q0, F) such that:
   – Q = Q1 x Q2
   – δ =
   – q0 =
   – F =

Correctness: We show that M recognizes L1 U L2:

Conclusion: A DFA recognizes L1 U L2, then L1 U L2 must be regular
Thm. The class of regular languages is closed under the union operation.

Proof:

Given: Two regular languages \( L_1, L_2 \), with DFAs \( M_1, M_2 \)

\(-\) \( M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1) \)
\(-\) \( M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2) \)

We construct a DFA \( M = (Q, \Sigma, \delta, q_0, F) \), such that:

\(-\) \( Q = Q_1 \times Q_2 \)
\(-\) \( \delta = \)
\(-\) \( q_0 = \)
\(-\) \( F = \)

Correctness: We show that \( M \) recognizes \( L_1 \cup L_2 \):

Conclusion: A DFA recognizes \( L_1 \cup L_2 \), then \( L_1 \cup L_2 \) must be regular

Which is true for \( q_0 \) in the proof?

A. \( q_0 = s_1 \)
B. \( q_0 = s_2 \)
C. \( q_0 = (s_1, s_2) \)
D. \( q_0 = \{(s_1, s_2)\} \)
Thm. The class of regular languages is closed under the union operation.

Proof:

Given: Two regular languages $L_1$, $L_2$, with DFAs $M_1$, $M_2$
- $M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$
- $M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$

We construct a DFA $M = (Q, \Sigma, \delta, q_0, F)$ such that:
- $Q = Q_1 \times Q_2$
- $\delta =$
- $q_0 =$
- $F =$

Correctness: We show that $M$ recognizes $L_1 \cup L_2$:

Conclusion: A DFA recognizes $L_1 \cup L_2$, then $L_1 \cup L_2$ must be regular

Which is FALSE for $\delta$?

A. $\delta: Q_1 \times Q_2 \to Q$
B. $\delta: Q_1 \times Q_2 \times \Sigma \to Q$
C. $\delta((x,y),c) = (\delta_1(x,c), \delta_2(y,c))$, for $c$ in $\Sigma$ and $(x,y)$ in $Q$
D. None of the above
Thm. The class of regular languages is closed under the union operation.

Proof:

Given: Two regular languages \( L_1, L_2 \), with DFAs \( M_1, M_2 \)

- \( M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1) \)
- \( M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2) \)

We construct a DFA \( M = (Q, \Sigma, \delta, q_0, F) \) such that:

- \( Q = Q_1 \times Q_2 \)
- \( \delta = \)
- \( q_0 = \)
- \( F = \)

Which is true for \( F? \)

A. \( F = F_1 \times F_2 \)
B. \( F = F_1 \cup F_2 \)
C. \( F = \{(x, y) \in Q \mid x \in F_1 \text{ or } y \in F_2 \} \)
D. None of the above

Correctness: We show that \( M \) recognizes \( L_1 \cup L_2 \):

Conclusion: A DFA recognizes \( L_1 \cup L_2 \), then \( L_1 \cup L_2 \) must be regular
Thm. The class of regular languages is closed under the union operation.

• Proof:
  • Given: Two regular languages $L_1$, $L_2$.
  • Want to show: $L_1 \cup L_2$ is regular.
  • Because $L_1$ and $L_2$ are regular, we know there exist DFAs $M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ that recognize $L_1$ and $L_2$.
    We construct a DFA $M = (Q, \Sigma, \delta, q_0, F)$, s.t.:
      - $Q = Q_1 \times Q_2$
      - $\delta((x,y),c) = (\delta_1(x,c), \delta_2(y,c))$, for $c$ in $\Sigma$ and $(x,y)$ in $Q$
      - $q_0 = (s_1, s_2)$
      - $F = \{ (x,y) \mid x \in F_1 \text{ or } y \in F_2 \}$
  • Correctness: We show $M$ recognizes $L_1 \cup L_2$. Let $x$ be a string whose computation in $M$ ends in state $(q,r)$. If $x$ is accepted by $M$, then $q$ is in $F_1$ or $r$ is in $F_2$. Therefore $x$ is accepted by $M_1$ or $M_2$, so $x$ is in $L_1 \cup L_2$. If $x$ is not accepted by $M$, then $q$ is not in $F_1$ and $r$ is not in $F_2$, and $x$ is not accepted by either $M_1$ or $M_2$. Therefore $x$ is not in $L_1 \cup L_2$.
  • Conclusion: There is a DFA that recognizes $L_1 \cup L_2$, so $L_1 \cup L_2$ is regular, and the class of regular languages is closed under union.
Thm. The class of regular languages is closed under the union operation.

Proof: *(Comment: A regular proof structure!)*

**What do we know?**
- L1 and L2 are regular, so by definition, we know there exist DFAs $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$ that recognize L1 and L2.

**What do we want to show?**
- L1 $\cup$ L2 is regular. To be regular, there must be a DFA that recognizes L1 $\cup$ L2.

**We show there is such a DFA:**
- We construct a DFA $M = (Q, \Sigma, \delta, q_0, F)$, s.t.:
  - $Q =$
  - $\delta =$
  - $q_0 =$
  - $F =$
- Correctness: Show M recognizes L1 $\cup$ L2
- A DFA recognizes L1 $\cup$ L2, therefore L1 $\cup$ L2 is regular. Q.E.D.
Thm. The class of regular languages is closed under Intersection

• Proof:

• Given: Two regular languages $L_1$, $L_2$.

• Want to show: $L_1 \cap L_2$ is regular.

• Because $L_1$ and $L_2$ are regular, we know there exist DFAs $M_1$ and $M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ that recognize $L_1$ and $L_2$. We construct a DFA $M = (Q, \Sigma, \delta, q_0, F)$, s.t.:
  
  - $Q = Q_1 \times Q_2$
  - $\delta((x,y),c) = (\delta_1(x,c), \delta_2(y,c))$, for $c$ in $\Sigma$ and $(x,y)$ in $Q$
  - $q_0 = (s_1, s_2)$
  - $F = \{ (x,y) \in Q \mid x \in F_1 \text{ and } y \in F_2 \}$

• Correctness: We show that $M$ recognizes $L_1 \cap L_2$. Let $x$ be a string whose computation in $M$ ends in state $(q,r)$. If $x$ is accepted by $M$, then $q$ is in $F_1$ and $r$ is in $F_2$. Therefore $x$ is accepted by $M_1$ and $M_2$, so $x$ is in $L_1 \cap L_2$. If $x$ is a string not recognized by $M$, then either $q$ is not in $F_1$ or $r$ is not in $F_2$. Therefore, $x$ is not accepted by one of $M_1$ and $M_2$, and so $M$ is not in $L_1 \cap L_2$.

• Conclusion: A DFA recognizes $L_1 \cap L_2$, so $L_1 \cap L_2$ is regular, and the class of regular languages is closed under union. Q.E.D.
Tips for Writing Closure Proofs

• A closure proof provides an answer to the question, "If I have a regular language, and do [blah] to it, is the new language still regular?"

• **GIVEN:** Write down what is known and give names to each of them and their component parts so you can use them later.

• **WANT TO SHOW:**
  • Announce what you will prove and your plan for it.

• **CONSTRUCTION:**
  • Let $M' = (Q', \Sigma', \delta', q_0', F')$, where ...
  • The construction will depend on the problem.
  • Though *not part of the proof*, a description in English of what you are trying to do is often useful.

• **CORRECTNESS:**
  • Here you prove that you construction actually works.

• **CONCLUSION:**
  • Finish by stating what you have proved.
Highlights of DFA’s so far

• Input: a string of symbols, of any length, over the alphabet
• A DFA $M$ accepts a string $w$ if using $w$ as input to $M$ from the start state leads to a final (accept) state
• Designed to recognize language $L$
  – The set of strings that are accepted by DFA
  – Remaining strings are rejected by DFA
• To specify a DFA $M$, either:
  – As example: Draw its State Diagram, with exactly 1 outgoing edge for each member of $M$’s alphabet, or
  – In a proof: Define each component in 5-tuple ($Q, \Sigma, \delta, q_0, F$)
• A language $L$ is regular if there is some DFA $M$ that recognizes $L$.
• Regular languages are closed under $\cup$ and $\cap$ operations