Deduction

• Not in book

• From a set of facts, deduce something that must be true

L is a regular language
So there must exist a DFA M that recognizes it

w is a string in the alphabet of \( \Sigma \)
So there must exist an \( n \geq 0 \) such that \( w = w_1 \cdots w_n \) and \( w_i \in \Sigma \)

Prove that regular languages are closed under complementation

\( \exists \omega \in \Sigma^* \) such that \( L = \{ \omega \in \Sigma^* \mid \text{for all } \sigma \in \Sigma, \omega \sigma \notin L \} \)

Show something is equivalent to a different problem

Prove that regular languages are closed under complementation

Let \( L \) be any regular language
Sufficient to show that \( \bar{L} \) is regular
Sufficient to show there exists a DFA that recognizes \( \bar{L} \)

Closure Properties of Regular Languages

• Construction
• Contradiction
• Induction
• Deduction
Overview

- Deduction
- Construction
- Contradiction
- Induction
- Closure Properties of Regular Languages

How Much Detail?

- How is current line derived?
- You can label each line and show when is used in deriving new lines
- You don't have to be super detailed
- Or obvious, or just use the previous line, don't need to number them
- How is current line derived?

So there exists an $w$ such that $m = m_1 \cdot \ldots \cdot m_n$ and $w \in L$.

Or we don't need to say that $n$ is a natural number.

Or what happens if $n = 0$, then there are no $w_i$.
Regular Languages Closed under Complementation

Let \( L \) be any regular language.
Sufficient to show that \( \overline{L} \) is regular.
Sufficient to show there exists a DFA that recognizes \( \overline{L} \).

Since \( L \) is regular, there exists a DFA that recognizes it. Call it \( M \).

Let \( M = (Q, \Sigma, \delta, q_0, F) \).

Construct DFA \( M' = (Q, \Sigma, \delta, q_0, F') \) where \( F' = Q - F \).

Claim: \( L(M') = \overline{L} \).

Let \( w \in L \) so \( M \) accepts \( w \).
Let \( q \) be the state that \( M \) is in at the end of processing \( w \).
So \( q \in F \).
So when \( M' \) processes \( w \), at the end of processing \( w \), it will also be in \( q \).
Since \( q \not\in F' \), \( M' \) does not accept \( w \).
So \( w \not\in L(M') \).

Similarly, if \( w \in L(M') \), then \( w \not\in L \).

So \( L(M') = \overline{L} \).

Proof by Construction

- Often asked to prove that a particular type of object exists.
- Question usually along the lines of: 
  Prove that there exists ...
  Prove that for each \( n > 2 \), there exists ...
- Proof technique:
  - Prove by constructing an example
  - Prove by proving wp with an example
- Sometimes disguised as...
- Of demonstrating how to construct the object.

Proof by Construction
Overview

- Deduction
- Construction
  \[ \Rightarrow \]
- Induction
- Closure Properties of Regular Languages

How Much Detail?

- How much detail is needed?
  - Need to convince me that you know how to do the proof
  - Should be clear what you need to prove (and why)
- Do you need to show construction does as intended?
  - Should be clear what you need to prove (and why)
  - Need to convince me that you know how to do the proof

\[ I \not\in L \quad \text{so} \quad I = (\langle M \rangle) I \]\n
\[ I \not\in L \quad \text{so} \quad I \not\in (\langle M \rangle I) I \quad \text{and if} \quad m \in L \quad \text{then} \quad (\langle M \rangle I) I \not\in L \]

Could add:
- Notice the word 'let', as in 'let \( x \) be ...'
- Sometimes you can handwave
- If question merely specify to just give the construction

So \( L = \bar{M} \)
Let \( A = \{ a^i b^i \mid i \geq 0 \} \). Assume \( A \) is regular.

So there exists DFA \( M \) such that \( L(M) = A \).

Let \( n \) be the number of states that \( M \) has.

Let \( s = a^n b^n + 1 \). Obviously \( s \in L(M) \).

Let \( r_0 \ldots r_{2n+2} \) be the state sequence for accepting \( s \).

Now consider the state sequence from \( r_0 \) to \( r_n \) in which each \( r_k \) on input \( a \) transitions to \( r_{k+1} \) for \( k \leq n \) (i.e., \( \delta(r_k, a) = r_{k+1} \)).

Since \( M \) just has \( n \) states, there must be at least one duplicate.

Let \( r_i \) and \( r_{i+j} \) be duplicate states, with \( j > 0 \), so \( r_i = r_{i+j} \).

Since \( r_{i+j} + 1 = \delta(r_{i+j}, a) \), then \( r_{i+j} + 1 = \delta(r_i, a) \).

So, \( r_0 \ldots r_i r_{i+j} + 1 \ldots r_{2n+2} \) is a valid state sequence and accepts \( a^{n+1} b^{n+1} - j \), contradicting the assumption that \( A \) is regular.
Induction

• Prove that all Xs have a certain property where all Xs can be categorized in terms of some property P based on the natural numbers, such as:
  - number of states in a DFA
  - number of nodes in a graph
  - some variable that is restricted to the natural numbers

• Break problem into a base case and an induction step:
  - Base case: prove that P(1) is true
  - Induction step: prove that if P(i) is true for i ≥ 1 then so is P(i + 1)

• After proving both parts, you know that P(i) must be true for i ≥ 1.

Prove that all Xs have a certain property where all Xs can be categorized in terms of some property P based on the natural numbers.

Overview

• Deduction
• Construction
• Contradiction
• Induction

• Closure Properties of Regular Languages
  - Induction
  - Construction
Proof by Induction:

Base case:
prove for \( n = 1 \)

LHS = \( \sum_{i=1}^{n} i \)

RHS = \( \frac{n(n+1)}{2} \)

Since LHS = RHS, true for base case

\( \sum_{i=1}^{n} i = \frac{n(n+1)}{2} \)

• Many variations of this:
  - Starting at a number other than 1
  - Needing to prove two base cases \( P(1) \) and \( P(2) \)
  - Needing to assume \( P(j) \) is true for all \( j \leq i \) in order to prove \( P(i+1) \)
  - Stating a number other than 1

Continued
Overview

- Deduction
- Construction
- Contradiction
- Induction

⇒ Closure Properties of Regular Languages

Continued

Induction step:
Assume it is true for \( n \), prove it is true for \( n + 1 \).

So assuming \( \sum_{i=1}^{n} i = \frac{n(n+1)}{2} \):

\[
\begin{align*}
\sum_{i=1}^{n+1} i &= \sum_{i=1}^{n} i + (n+1) \\
&= \frac{n(n+1)}{2} + (n+1) \\
&= \frac{n^2 + 3n + 2}{2}
\end{align*}
\]

LHS = \( \frac{n^2 + 3n + 2}{2} \)

RHS = \( \frac{n^2 + 3n + 2}{2} \)

Since LHS = RHS, true for \( n + 1 \).

\[
\begin{align*}
(1) & \quad z/(z + u + c) = \\
(2) & \quad z/(z + u)(1 + u) = SHT \\
(3) & \quad z/(z + u + c) = \\
(4) & \quad z/(z + u)(1 + u) = \\
(5) & \quad (1 + u) + z/(1 + u) = \\
(6) & \quad (1 + u) + \sum_{i=1}^{n} i = SHT \\
(7) & \quad z/(z + u)(1 + u) = \sum_{i=1}^{n} i = \sum_{i=1}^{n} i \\
(8) & \quad \sum_{i=1}^{n} i = \sum_{i=1}^{n} i \\
(9) & \quad \sum_{i=1}^{n} i = \sum_{i=1}^{n} i
\end{align*}
\]
Proof Idea (continued)

Approach 1:
- We could have $M_1$ first simulate $M_1$ and then simulate $M_2$.
- This won't work because in simulating $M_1$, we will have used up the input with $w$ on it, and we won't be able to get it back.

Approach 2:
- Simulate both $M_1$ and $M_2$ at the same time!
- Let's assume $M_1$ be the product of states of $M_1$ and $M_2$.
- This won't work because in simulating $M_1$, we will have used up the input with $w$. We could have first simulated $M_1$ and then simulated $M_2$.

Closed under Union

Definition of Union:

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

Class of regular languages is closed under union.

Let $A$ and $B$ be regular languages (each accepted by a FA), then there is a FA $M$ that accepts $A \cup B$.

Proof by construction:

If $A$ and $B$ are regular languages, there exists FA $M_1$ and $M_2$ such that $L(M_1) = A$ and $L(M_2) = B$.

Let $M' = \langle Q', \Sigma, \delta', q_0', F' \rangle$ be a FA that accepts $A \cup B$. Since $A$ and $B$ are regular languages, there exists FA $M_1$ and $M_2$ such that $L(M_1) = A$ and $L(M_2) = B$. Then $M'$ is a FA that accepts $A \cup B$.
Example Continued

• What are the states of $M$ (draw them in a 3x2 array)?

- Where does $M$ transition to from $a$ on $0$?

- Where does $M$ transition to from $a$ on $1$?

- Take each state, like $a$, and each input, like $0$, and ask: Where are the transitions of $M$?

• What are the states of $M$ (draw them in a 3x2 array)?

Example

Let $\Sigma = \{0, 1\}$

Let $A_1$ be strings in which the number of 0's is divisible by 3

- Draw a state diagram for $M_1$ with 3 states: $a, b, c$

- Draw the 3 states horizontally

Let $A_2$ be strings that have an even number of 1s

- Draw a state diagram for $M_1$ with 2 states: $d, e$

- Draw the two states vertically

Let $A_3$ be strings that have an even number of 1s

- Draw the two states vertically
Other Operations

Regular Languages

- Intersection
\[ A \cap B = \{ x \mid x \in A \text{ and } x \in B \} \]
- Complementation
\[ \overline{A} = \{ x \mid x \notin A \} \]
- Concatenation
\[ A \circ B = \{ xy \mid x \in A \text{ and } y \in B \} \]
- Star
\[ A^* = \{ x_1 x_2 \cdots x_k \mid k \geq 0 \text{ and each } x_i \in A \} \]