ANALYSIS AND DESIGN OF ALGORITHMS

Module 02: Introduction to Theoretical Analysis

Ammar Sherif

Review

- Categories of Analysis
- Empirical Analysis
- Recursion
- Fibonacci
- "Sapere Aude"

Outline

- Propositions and Quantifiers
- Implication and De Morgan Laws
- Proofs
  - Direct
  - Contradiction
  - Contrapositive
  - Induction and Strong Induction
- Correctness Proofs
- O, Ω, Θ

LOGIC

Why to study Logic and Proofs

CORRECTNESS

MAKE PROBLEM EASIER

EXPLAIN WHY THE ALGORITHM TOOK A DECISION

FUNNY

Propositions

- Statements have truth values [True/False]
- Examples: "This is a time machine," or "this is a cat"
- What is not a proposition?
  - Questions
  - Orders
  - Statements include contradiction
- Logic operators: not (¬), and (∧), or (∨)
- Predicates
  - $P(x)$: $x > 2$
  - $P(1) \equiv False$
  - $P(3) \equiv True$

Quantifiers

- There **exists**, symbol: $\exists$
- Example: "There is an integer greater than 5"

$$\exists_{x \in \mathbb{Z}} x > 5$$

- **Universal, for all**, symbol: $\forall$
- Example: "for all integers, n+1 is greater than n"

$$\forall_{n \in \mathbb{Z}} \; n + 1 > n$$

De Morgan laws

- “ He is a student and he takes 304” $(A \wedge B)$
- **Negation**: “He is not a student, **or** he does not take 304”
$$\overline{A \wedge B} \equiv \neg(A \wedge B) \equiv \overline{\boldsymbol{A}} \vee \overline{\boldsymbol{B}}$$
- “It rains, or it is sunny” $(A \vee B)$
- **Negation**: “It does not rain, **and** it is not sunny”
$$\overline{A \vee B} \equiv \overline{\boldsymbol{A}} \wedge \overline{\boldsymbol{B}}$$

De Morgan laws with Quantifiers

- "All integers are positive;" its *negation* would be "There is **at least** one non-positive integer"
- Proof:

$$\neg(\forall_{x \in \mathbb{Z}}\, x > 0) \equiv \overline{\big((0 > 0) \wedge (1 > 0) \wedge (-1 > 0) \wedge \cdots\big)}$$
$$\equiv \overline{(\cdots \wedge P(-2) \wedge P(-1) \wedge P(0) \wedge P(1) \wedge \cdots)}$$
$$\equiv \big(\cdots \vee \overline{P(-2)} \vee \overline{P(-1)} \vee \overline{P(0)} \vee \overline{P(1)} \vee \cdots\big) \equiv \exists_{\boldsymbol{x} \in \mathbb{Z}}\, \overline{\boldsymbol{P(x)}}$$

- "There is an integer greater than 2 that is even and prime;" its negation would be "All integers greater than 2 are not even and prime at the same time."
- Proof:

$$\neg\big(\exists_{x>2}\, P(x)\big) \equiv \overline{(P(3)\ \vee P(4) \vee P(5) \vee \cdots)} \equiv \big(\overline{P(3)} \wedge \overline{P(4)} \wedge \overline{P(5)} \wedge \cdots\big)$$
$$\equiv \forall_{x>2} \overline{\boldsymbol{P(x)}}\ \not\equiv \forall_{x\leq 2} \overline{P(x)} \not\equiv \forall_{x\leq 2}\, P(x)$$

A	B	$A \Longrightarrow B$
T	T	T
T	F	F
F	T	T
F	F	T

Implication rule

- "if A is true, then B is true"
- $A \Longrightarrow B$

PROOFS

Direct Proofs

- **Required**: $\boldsymbol{P \Longrightarrow Q}$
- $P \Longrightarrow Q_1 \Longrightarrow Q_2 \Longrightarrow \cdots \Longrightarrow Q$
- Example: if $x^2 - 25 = 0 \Longrightarrow \boldsymbol{x = \pm 5}$
- **Proof**:

$$x^2 - 25 = 0 \Longrightarrow (x-5)(x+5) = 0 \Longrightarrow$$
$$(x = 5) \vee (x = -5) \Longrightarrow \boldsymbol{x = \pm 5}$$

Proof by Contrapositive

- $(\boldsymbol{P \Longrightarrow Q}) \Leftrightarrow (\boldsymbol{\bar{Q} \Longrightarrow \bar{P}})$
- **Intuition**: "if he is a human, he has a brain."
- **Example**: Assuming $n, a, b \in \mathbb{Z}$, prove $n \nmid ab \Longrightarrow (n \nmid a \wedge n \nmid b)$
- $P \coloneqq n \nmid ab$ and $Q \coloneqq (n \nmid a \wedge n \nmid b)$
- **Proof**: $\bar{Q} \equiv \left(\overline{n \nmid a} \vee \overline{n \nmid b}\right) \Longrightarrow n \mid a \vee n \mid b \Longrightarrow$
  - $n \mid a \Longrightarrow a = nk \Longrightarrow ab = n(kb) \Longrightarrow n \mid ab \Longrightarrow \boldsymbol{\bar{P}}$
  - $n \mid b \Longrightarrow b = nt \Longrightarrow ab = n(ta) \Longrightarrow n \mid ab \Longrightarrow \boldsymbol{\bar{P}}$
- $(\bar{Q} \Longrightarrow \bar{P}) \Longrightarrow (\boldsymbol{P \Longrightarrow Q})$

Proof by Contradiction

- $\overline{\boldsymbol{P}} \Longrightarrow \boldsymbol{False}$
- **Intuition**: assume your theorem is *false*, prove this assumption leads to a contradiction.
- **Example**: prove that $\sqrt{2}$ is irrational, P
- **Proof**: Assume P is false that is $\sqrt{2}$ is rational $\Longrightarrow \exists_{a,b\in\mathbb{Z}^+}\ \sqrt{2} = \frac{a}{b}$.
  Without loss of generality, assume the $\gcd(a, b) = 1$.
  $\Longrightarrow a^2 = 2b^2 \Longrightarrow a^2$ is even $\Longrightarrow a$ is even $\Longrightarrow a^2 = (2k)^2 = 4k^2$ where $k < a \Longrightarrow a^2 = (4k^2 = 2b^2)$
  $\Longrightarrow 2k^2 = b^2 \Longrightarrow b^2$ is even $\Longrightarrow (b = 2n)$ where $n < b \Longrightarrow \boldsymbol{\sqrt{2}} = \dfrac{a}{b} = \dfrac{2k}{2n} = \boldsymbol{\dfrac{k}{n}}$ which is in a lower terms
  $\Longrightarrow \boldsymbol{\gcd(a, b) \neq 1}$ [Contradiction]

Proof by Induction

- $\boldsymbol{P(a_0) \wedge \left(P(a_i) \Rightarrow P(a_{i+1})\right)}$
- **Intuition**: dominoes
- **Example**: prove

$$P(n): \sum_{i=0}^{n} i = \frac{n(n+1)}{2}$$

- **Base case**: $P(0): \sum_{i=0}^{0} i = 0, \frac{0 \times 1}{2} = 0 \Rightarrow \boldsymbol{P(0)}$ is True
- **Inductive step**: Assume $P(n)$ is True $\Rightarrow$

$$\sum_{i=0}^{n+1} i = (n+1) + \sum_{i=0}^{n} i = (n+1) + \frac{n(n+1)}{2} = \frac{2(n+1) + n(n+1)}{2} = \boldsymbol{\frac{(n+1)(n+2)}{2}} \Rightarrow \boldsymbol{P(n+1)}$$

- $P(0) \Rightarrow P(1) \Rightarrow P(2) \Rightarrow \cdots \Rightarrow \boldsymbol{P(n)}$

Proof by Strong Induction

- $\boldsymbol{P(a_0) \wedge \Big(\big(P(a_0) \wedge \cdots \wedge P(a_i)\big) \Rightarrow P(a_{i+1})\Big)}$
- **Example**: prove prime factorization

  $\mathrm{P}(n)\colon\ \forall_{n\geq 2}\ n = p_0 p_1 \cdots p_k$ where p_i is **prime**

  $6 = 2 \times 3$
- **Base case**: $P(2) = 2$ "prime" $\Rightarrow True$
- **Inductive step**: Assume $P(2) \wedge \cdots \wedge P(n) \equiv True \Rightarrow n + 1$ has two cases
  - Case 1: $n + 1$ itself is prime $\Rightarrow \boldsymbol{P(n+1)}$ is $True$
  - Case 2: $n + 1 = a \times b$, where $2 \leq a, b < (n + 1) \Rightarrow a = p_1 p_2 \cdots p_k$ from $P(a) \wedge b = q_1 q_2 \cdots q_m$ from $P(b) \Rightarrow$
    $n + 1 = a \times b = p_1 p_2 \cdots p_k q_1 q_2 \cdots q_m$ where p_i and q_j are primes $\Rightarrow \boldsymbol{P(n+1)}$ is $True$
- $P(2) \Rightarrow P(3),\ P(2) \wedge P(3) \Rightarrow P(4),\ P(2) \wedge P(3) \wedge P(4) \Rightarrow P(5), \cdots, P(2) \wedge \cdots \wedge P(n-1) \Rightarrow \boldsymbol{P(n)}$

CORRECTNESS PROOFS

Remarks and Steps

- We can **only** prove "Statements," "Propositions," or "**Hypothesis**"
  - Therefore, the first step is to formulate such "Hypothesis"
- Guidelines for the Hypothesis
  - It differs from algorithm to another.
  - It should include an answer to what do you mean by the algorithm is "**correct**"?
  - To validate your statement, *assume* that you have proved the statement, and make sure with that your algorithm becomes true. [If not, **redfine** your statement until proven]
  - Example for a sorting algorithm, the statement might be
    "By the end of the algorithm, $A[0] \leq A[1] \leq \cdots \leq A[n-1]$"

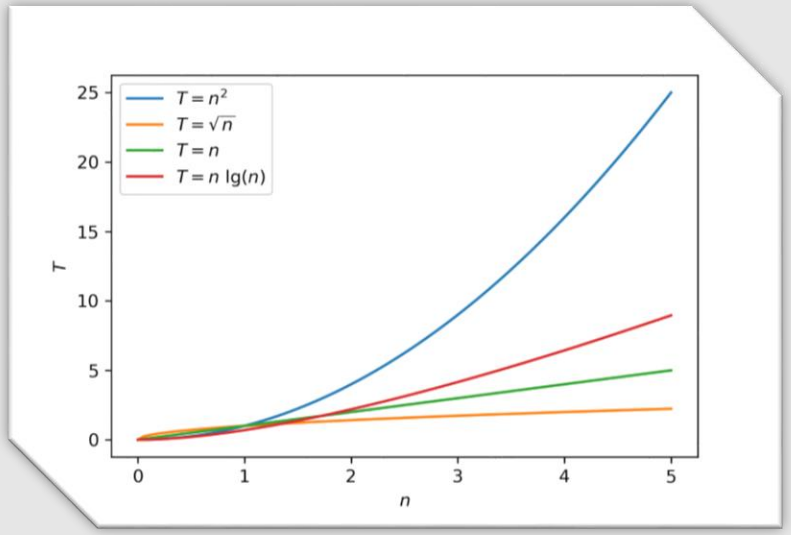
Remarks and Steps (Cont.)

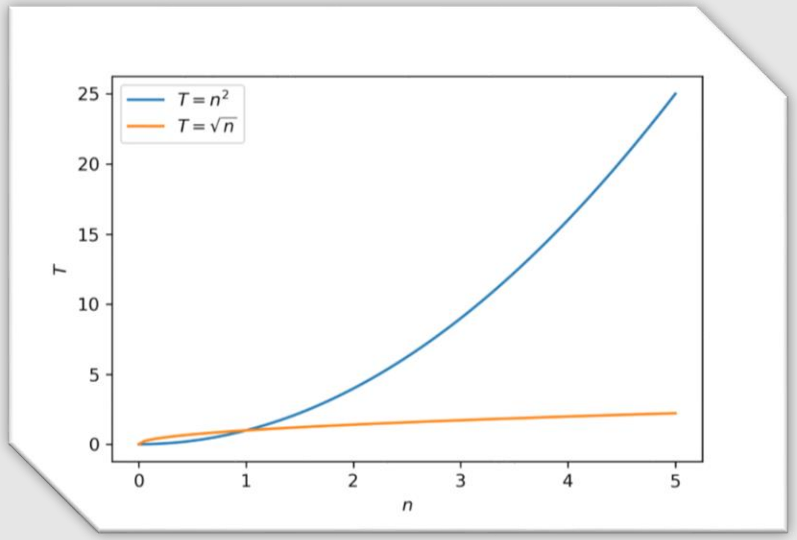
- Guidelines for the Hypothesis:
  - Example for a sorting algorithm, the statement might be "By the end of the algorithm, $A[0] \leq A[1] \leq \cdots \leq A[n-1]$"
  - When using induction, there must be some variable to induct upon "By the end of the $\boldsymbol{i^{th}}$ iteration, $A[0] \leq A[1] \leq \cdots \leq A[i-1]$"
- If you use inductive scheme, do not forget its guidelines
  - Base Case(s) and the Inductive Step
  - Base case depends upon the range of the inductive variable: $n \leq 5,\ n \geq 0$
  - Within the inductive step, start with $H(k)$, and try to prove the correctness of $H(k+1)$
  - Quite straightforward, you go with the code, apply each step. Then, you try to get $H(k+1)$
- Check out the next example in Jupyter Notebook

ALGORITHMIC COMPLEXITY

Growth rates

- To achieve machine-independence, compare between growth rate instead of absolute values, for large enough input size.
- Example: $T = c_1 * n,\ T = c_2 * n$
  We want to say T grows **linearly** with n
- Divide functions into categories depending on their growth
- Develop notation **upper**, **lower**, and **tight** bounds





Big-Oh the upper bound

- A notation for **upper bounds**; the performance will never be worse than, will be at most.
- Formally:

$$O\big(g(n)\big) = \big\{f(n)\!: \exists_{c,k}\ \forall_{n \geq k}\ f(n) \leq c.\, g(n)\ \big\}$$

- $\leq$
- Examples:
  - $n \in O(n^2),\ n = O(n^2)$
  - $10^5 = O(1)$
  - $10^3\boldsymbol{n^2} + 10^6 n + 10^9 = O\big(\boldsymbol{n^2}\big)$
  - $n^2 = O(2n^2)$

Big-Omega the lower bound

- A notation for **lower bounds**; the performance will never be better than, will be at least.
- Formally:

$$\Omega\big(g(n)\big) = \big\{f(n)\colon \exists_{c,k}\ \forall_{n\geq k}\ f(n) \geq c.\, g(n)\,\big\}$$

- $\geq$
- Examples:
  - $n \lg(n) = \Omega(n)$
  - $e^n = \Omega(n^3) \Longrightarrow n^3 = O(e^n)$

Big-Theta the tight bound

- A notation for **tight bounds**; the performance will almost be equivalent.
- Formally:
  $\Theta(g(n)) = \{f(n) \colon \exists_{c_1,c_2,k} \forall_{n \geq k}\ c_1 g(n) \leq f(n) \leq c_2 g(n)\}$
  $= O(g(n)) \cap \Omega(g(n))$
- $=$
- Example:
  - $10n = \Theta(n)$

Little-oh the strict upper bound

- A notation for an **upper bound but not tight**; the performance will never be as bad as.
- Formally:

$$\mathrm{o}\big(g(n)\big) = \{f(n) \colon \forall_{c>0} \exists_k \, \forall_{n \geq k} \quad f(n) < cg(n) \,\}$$

- <
- Example:
  - $10n = \mathrm{o}(n^2) \neq o(n)$

Little-omega the strict lower bound

- A notation for a **lower bound but not tight**; the performance will never be as good as.
- Formally:

$$\omega\big(g(n)\big) = \{f(n): \forall_{c>0} \exists_k \ \forall_{n \geq k} \ \ f(n) > cg(n)\}$$

- $>$
- Example:
  - $10n = \omega(\sqrt{n}) \neq \omega(n)$

Recap

- Propositions and Quantifiers
- Implication and De Morgan Laws
- Proofs
  - Direct
  - Contradiction
  - Contrapositive
  - Induction and Strong Induction
- Correctness Proofs
- Ω, Θ, O

References

- Discrete Mathematics and its Applications by Rosen
- Introduction to Algorithms by Coreman, Leiserson, Rivest, and Stein (CLRS)

QUESTIONS

Contacts

- Github: Tutorials Repo
- Email ID: Asherif