COMP 182 Algorithmic Thinking

Sets, Propositional Logic, Predicates, and Quantifiers

Luay Nakhleh
Computer Science
Rice University

Reading Material

- Chapter 1, Sections 1, 4, 5
- Chapter 2, Sections 1, 2

- Mathematics is about statements that are either **true** or **false**.
- Such statements are called **propositions**.
- We use **logic** to describe them, and **proof techniques** to prove whether they are true or false.

Propositions

- $5>7$
- The square root of 2 is irrational.
- A graph is bipartite if and only if it doesn’t have a cycle of odd length.
- For $n>1$, the sum of the numbers $1,2,3,\ldots,n$ is n^2.

Propositions?

- $E=mc^2$
- The sun rises from the East every day.
- All species on Earth evolved from a common ancestor.
- God does not exist.
- Everyone eventually dies.

- And some of you might already be wondering: “If I wanted to study mathematics, I would have majored in Math. I came here to study computer science.”

- Computer Science ***is*** mathematics, but we almost exclusively focus on aspects of mathematics that relate to computation (that can be implemented in software and/or hardware).

- Logic ***is*** the language of computer science and, mathematics ***is*** the computer scientist's most essential toolbox.

Examples of "CS-relevant" Math

- Algorithm A correctly solves problem P.
- Algorithm A has a worst-case running time of $O(n^3)$.
- Problem P has no solution.
- Using comparison between two elements as the basic operation, we cannot sort a list of n elements in less than $O(n \log n)$ time.
- Problem A is NP-Complete.

- “Algorithm A is correct” is a proposition that requires a mathematical proof.
  - All students in the course thinking that it is true is not a proof.
  - Showing it is true on 1 million examples is not a proof.

- “Problem P has no solution” is a proposition that requires a mathematical proof.
  - Your inability to come up with a solution to Problem P is not a proof that a solution doesn’t exist.
  - All your 5,000 Facebook friends not being able to come up with a solution doesn’t make the statement true either.

- Despite decades of work by so many brilliant researchers, no one has been able to come up with a polynomial-time algorithm for the Traveling Salesman Problem (TSP).
- Still, no computer scientist or mathematician would state “TSP has no polynomial-time solution” because such a statement would require a mathematical proof and such a proof has not been found yet.

- It is important to note that decades of work by brilliant researchers not resulting in a polynomial-time algorithm for TSP do strengthen our ***<u>belief</u>*** that the ***<u>conjecture</u>*** that "TSP has no polynomial-time solution" is true.
- This belief could, for example, direct other brilliant researchers to focus on proving the conjecture true (rather than false).
- However, no matter how strong our belief is, it is still not a proof.

Sets

- A <u>set</u> is an unordered collection of items.
- We write $a \in S$ to denote that a is an <u>element of</u> set S, or that set S <u>contains</u> element a.
- <u>Roster method</u> description of sets: B={0,1}, C={a,b,c,d}, D={#,$,%,&,@}
- <u>Set builder or set comprehension</u> description of sets: $F=\{x \mid x \text{ is an odd integer}\}$, $G=\{y \mid y \text{ is an integer that is divisible by } 7\}$

Sets

- An element of a set cannot appear more than once in the set.
- For example, $\{a,b,b,c\}$ is *not* a set.
- A mathematical structure that allows for an element to appear more than once is called <u>multiset</u> or <u>bag</u>. In this course, we will only work with sets.

Special Sets

- The set of natural numbers $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$
- The set of integers $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$
- The set of positive integers $\mathbb{Z}^+ = \{1, 2, \ldots\}$
- The set of rational numbers $\mathbb{Q} = \{\frac{p}{q} | p, q \in \mathbb{Z}, q \neq 0\}$
- The set of real numbers $\mathbb{R}$
- The set of positive real numbers $\mathbb{R}^+$

The Empty Set

- The <u>empty set</u> is the set that contains no elements.
- Denoted by $\varnothing$ or $\{\}$.
- Important: The set $\{\varnothing\}$ is *not* empty. Rather, it is a set that contains one element that is $\varnothing$.

Cardinality of Sets

- The cardinality of a finite set S, denoted by $|S|$, is the number of elements in S.
- $|\{a,b,c\}| = 3$
- $|\varnothing| = 0$
- $|\{\varnothing\}| = 1$

Cardinality of Sets

- Not all sets are finite.
- Infinite sets can be <u>countable</u> or <u>uncountable</u>.
- More on this later in the semester.

Propositional Logic

Propositions

- A proposition is a declarative sentence that is either true or false, but not both.
- We use propositional variables (e.g., p, q, r, s,…) to represent propositions.

Propositions

- Propositions:
  - $3 \in \{1,2,4\}$
  - $|\{0,1\}|=2$
  - $7 \notin \{a,b,c\}$
- Not propositions:
  - $1+1$
  - $\{a,b,c\}$
  - $|\{5,12,19\}|$

Compound Propositions

- If p is a proposition, $\neg p$ is its <u>negation</u>.
- If p and q are two propositions, then
  - $p \wedge q$ ("p and q") is their <u>conjunction</u>
  - $p \vee q$ ("p or q") is their <u>disjunction</u>

Truth Values

- The <u>truth value</u> of a proposition is true, denoted by T, if it is a true proposition, and the truth value is false, denoted by F, if it is a false proposition.
- True propositions:
  - $|\{a,b\}|=2$ $|\emptyset|<|\{1\}|$ $7\notin\{1,5,9,12\}$
- False propositions:
  - $|\{\emptyset\}|=0$ $7\in\{1,5,9,12\}$

Truth Table

- For a compound proposition, one way to determine the truth value of the proposition is by using a truth table.
- The truth table has one row for each combination of T and F for the primitive propositions.

Truth Table

TABLE 1 **The Truth Table for the Negation of a Proposition.**

p	$\neg p$
T	F
F	T

TABLE 2 **The Truth Table for the Conjunction of Two Propositions.**

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

TABLE 3 **The Truth Table for the Disjunction of Two Propositions.**

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

XOR, If, and Iff

TABLE 4 **The Truth Table for the Exclusive Or of Two Propositions.**

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

TABLE 5 **The Truth Table for the Conditional Statement $p \rightarrow q$.**

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

TABLE 6 **The Truth Table for the Biconditional $p \leftrightarrow q$.**

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Truth Tables of Compound Propositions

TABLE 7 **The Truth Table of $(p \vee \neg q) \rightarrow (p \wedge q)$.**

p	q	$\neg q$	$p \vee \neg q$	$p \wedge q$	$(p \vee \neg q) \rightarrow (p \wedge q)$
T	T	F	T	T	T
T	F	T	T	F	F
F	T	F	F	F	T
F	F	T	T	F	F

Tautology

- A tautology is a compound proposition whose truth value is T under all truth assignments to its propositional variables.
- Examples:
  - $p \vee \neg p$
  - $(p \rightarrow q) \vee \neg q$
  - $\mathrm{T} \vee p$

Contradiction

- A <u>contradiction</u> is a compound proposition whose truth value is F under all truth assignments to its propositional variables.
- Examples:
  - $p \wedge \neg p$
  - $F \wedge p$

Propositional Satisfiability

- A compound proposition is <u>satisfiable</u> if it is not a contradiction.
- A truth assignment to the propositional variables that make the compound proposition T is called a <u>solution</u> of this particular satisfiability problem.

Propositional Satisfiability

- The Propositional Satisfiability Problem, commonly known as SAT, is a decision problem that plays a central role in computer science.
- It is defined as:
  - Input: A compound proposition φ
  - Output: "Yes", if φ is satisfiable, and "No" otherwise

Propositional Satisfiability

- A trivial algorithm for solving SAT would build the truth table of φ and check the rightmost column for a T.
- If φ has n propositional variables, how many rows does the truth table have?
- If the computer can build and evaluate *1000* rows a second, how many seconds does this algorithm take if $n=10$? If $n=1000$?

Propositional Satisfiability

- Two seminal results:
  - Stephen Cook (1971) showed that SAT is NP-Complete.
  - Richard Karp (1972) introduced polynomial-time reductions as a tool to show other problems are NP-Complete.
- Both Cook and Karp won the Turing award for these contributions.

Predicate Logic

Predicate Logic

- In mathematics and computer science, we often find statements that involve variables.
- For example, $x \in \{1,2,3\}$.
- In this example, x is the variable, and "$x \in \{1,2,3\}$" is the predicate.

Predicate Logic

- The statement $x \in \{1,2,3\}$ can be denoted by <u>propositional function</u> $P(x)$.
- This propositional function evaluates to either T or F once a value has been assigned to variable x, in which case the statement $P(x)$ becomes a proposition.
- What is $P(1)$? $P(3)$? $P(7)$?

Predicate Logic

- These statements and functions may involve any number of variables.
- For example, $Q(x,y)$ is the statement "$x \in \{1,2,3\} \wedge y \notin \{a\}$".
- What is the value of $Q(1,a)$? $Q(1,b)$?

Quantifiers

- Another way to turn a propositional function into a proposition is via <u>quantification</u>.
- <u>Predicate calculus</u> is the area of logic that deals with predicates and quantifiers.

Universal Quantification

- The <u>universal quantification</u> of $P(x)$ is the statement "$P(x)$ for all values of x in the domain of discourse" and is denoted by $\forall x P(x)$.

Existential Quantification

- The <u>existential quantification</u> of $P(x)$ is the statement "$P(x)$ for some value of x in the domain of discourse" and is denoted by $\exists x P(x)$.

Quantifiers

Statement	***When True?***	***When False?***
$\forall x\, P(x)$	$P(x)$ is true for every x.	There is an x for which $P(x)$ is false.
$\exists x\, P(x)$	There is an x for which $P(x)$ is true.	$P(x)$ is false for every x.

Negating Quantified Expressions

TABLE 2 **De Morgan's Laws for Quantifiers.**

Negation	*Equivalent Statement*	*When Is Negation True?*	*When False?*
$\neg\exists x P(x)$	$\forall x \neg P(x)$	For every x, $P(x)$ is false.	There is an x for which $P(x)$ is true.
$\neg\forall x P(x)$	$\exists x \neg P(x)$	There is an x for which $P(x)$ is false.	$P(x)$ is true for every x.

Quantifiers

- What are the truth values of the following statements if the domain consists of all integers:

a) $\forall n(n^2 \geq 0)$ **b)** $\exists n(n^2 = 2)$

c) $\forall n(n^2 \geq n)$ **d)** $\exists n(n^2 < 0)$

Nested Quantifiers

Statement	*When True?*	*When False?*
$\forall x \forall y P(x, y)$ $\forall y \forall x P(x, y)$	$P(x, y)$ is true for every pair x, y.	There is a pair x, y for which $P(x, y)$ is false.
$\forall x \exists y P(x, y)$	For every x there is a y for which $P(x, y)$ is true.	There is an x such that $P(x, y)$ is false for every y.
$\exists x \forall y P(x, y)$	There is an x for which $P(x, y)$ is true for every y.	For every x there is a y for which $P(x, y)$ is false.
$\exists x \exists y P(x, y)$ $\exists y \exists x P(x, y)$	There is a pair x, y for which $P(x, y)$ is true.	$P(x, y)$ is false for every pair x, y.

Negating Nested Quantifiers

- $\neg \forall x \exists y \exists z \forall w P(x,y,z,w)$
- $\exists x \forall y \forall z \exists w \neg P(x,y,z,w)$

Nested Quantifiers

- What are the truth values of the following statements if the domain consists of all integers:

a) $\forall n \exists m (n^2 < m)$

b) $\exists n \forall m (n < m^2)$

c) $\forall n \exists m (n + m = 0)$

d) $\exists n \forall m (nm = m)$

Back to Sets

Subsets

- Let A and B be two sets.
- A is a subset of B, denoted by $A \subseteq B$, if the following quantified expression is true:

$$\forall x (x \in A \rightarrow x \in B)$$

Proper Subsets

- Let A and B be two sets.
- A is a <u>proper subset</u> of B, denoted by $A \subset B$, if the following quantified expression is true:

$$\forall x(x \in A \rightarrow x \in B) \land \exists x(x \in B \land x \notin A)$$

Equal Sets

- Let A and B be two sets.
- A and B are <u>equal</u>, denoted by $A{=}B$, if the following quantified expression is true:

$$\forall x(x \in A \leftrightarrow x \in B)$$

Power Sets

- The <u>power set</u> of set *A*, denoted by *P(A)* or 2^A, is the set of all subsets of *A*.
- What is the power set of *{1,2,3}*? Of ∅?

Set Operations

- Union: $A \cup B = \{x | x \in A \vee x \in B\}$
- Intersection: $A \cap B = \{x | x \in A \wedge x \in B\}$
- Difference: $A \setminus B = \{x | x \in A \wedge x \notin B\}$
- Complement: $\overline{A} = \{x \in U | x \notin A\}$
- Cartesian product: $A \times B = \{(x, y) | x \in A \wedge y \in B\}$

Set Identities

Identity	*Name*
$A \cap U = A$ $A \cup \emptyset = A$	Identity laws
$A \cup U = U$ $A \cap \emptyset = \emptyset$	Domination laws
$A \cup A = A$ $A \cap A = A$	Idempotent laws
$\overline{(\overline{A})} = A$	Complementation law
$A \cup B = B \cup A$ $A \cap B = B \cap A$	Commutative laws
$A \cup (B \cup C) = (A \cup B) \cup C$ $A \cap (B \cap C) = (A \cap B) \cap C$	Associative laws
$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	Distributive laws
$\overline{A \cap B} = \overline{A} \cup \overline{B}$ $\overline{A \cup B} = \overline{A} \cap \overline{B}$	De Morgan's laws
$A \cup (A \cap B) = A$ $A \cap (A \cup B) = A$	Absorption laws
$A \cup \overline{A} = U$ $A \cap \overline{A} = \emptyset$	Complement laws

Questions?