1 # CS3203 #2

5/26/04

Janak J Parekh

2 # Administrivia

- Textbooks delayed. ☹
  - Should I delay the homework?
- No class next Monday – Memorial Day

3 # Proofs

- **Theorem**: Statement that can be shown true via a **proof**.
  - **Conjecture** is a statement whose truth value is unknown; turns into a theorem given a proof
- **Asiom**/**postulate** are underlying assumptions about mathematical structures, hypothesis, and previously proved theorems
- **Rules of inference** tie steps together
- Need to avoid **fallacies**
- **Lemma**: mini-proof used in other proofs; a **corollary** is a “side-effect” of a proof.

4 # Rules of inference

- Need these for proofs
- **Modus ponens**, or **law of detachment**
  - Example: consider the tautology
    $(p \land (p \rightarrow q)) \rightarrow q$
  - Either *p* is true, in which case $p \rightarrow q$ depends on q, or *p* is false, in which case $p \rightarrow q$ is always true
  - Therefore, this is equivalent to q
  - Other rules – see page 58 and 60
- Multiple ways of writing tautologies…

5 # Valid argument

- An argument is **valid** if all the hypotheses are true.
- Valid doesn’t mean true!
  - All the propositions must be true
- Scenario:
  - It is not sunny this afternoon and it is colder than yesterday.
  - We will go swimming only if it is sunny.
  - If we do not go swimming, then we will take a canoe trip.
  - If we take a canoe trip, then we will be home by sunset.
  - Conclusion: we will be home by sunset.

6 # Fallacy

- An invalid argument
- **Fallacy of affirming the conclusion**: $[(p \rightarrow q) \land q] \rightarrow p$
  - Just because q is true, doesn’t mean p is
  - “If you do homework, then you are smart”; “you are smart”; “therefore you did homework” doesn’t fly, i.e., homework isn’t the only criterion for becoming smart.
- **Fallacy of denying the hypothesis:** $[(p \rightarrow q) \land \neg p] \rightarrow \neg p$

7 # Rules of inference with quantifiers

- **Universal instantiation**: given $\forall x P(x)$, we can conclude P(c).
- **Universal generalization:** given P(c) true for all c, we can say $\forall x P(x)$ by selecting a truly arbitrary c.
- **Existential instantiation**: If $\exists x P(x)$, *select an appropriate c* for which P(c). We bind “c” to it and use it through the argument.
- **Existential generalization:** If P(c) is known for a c, we can state $\exists x P(x)$ is true.

- Example:
  - "Everyone in this discrete mathematics class has taken a course in Computer Science"
  - "John is a student in this class"
  - implies "John has taken a course in Computer Science."
- Mathematical theorems often omit the universal quantifier (i.e., for all real numbers, etc.) – it's all done implicitly.

8 Methods of proving theorems

- **Direct proof:** what we've been doing so far
  - Assume p is true and use rules of inference to show that q must be true
  - Example: If n is an odd integer, then n^2 is an odd integer
    - Definition 1: n is even if k such that $n = 2k$ and odd if k such that $n = 2k+1$
- **Indirect proof:** use contrapositive
  - Show that if q is false, p must be false
  - Example: If 3n+2 is odd, then n is odd → assume n is even.
- **Vacuous proof:** if the hypothesis *p* is false, then p → q is automatically true
  - Example: P(0) where P(n) "If n > 1, n+1 > 1."

9 Proving theorems (II)

- **Proof by contradiction:** show that ¬p → q is true, i.e., ¬p → F or q = F. Therefore, ¬p must be false and p must be true.
  - Example: Show at least 4 of any 22 days must fall on the same day of the week => assume this is false
- **Proof by cases:** decouple $(p_1 \vee p_2 \vee \ldots \vee p_n) \rightarrow q$ into $(p_1 \rightarrow q) \wedge (p_2 \rightarrow q) \wedge (p_n \rightarrow q)$.
- **Proofs of equivalence:** decouple $p \leftrightarrow q$ into $(p \rightarrow q) \wedge (q \rightarrow p)$

10 Mistakes in proofs, techniques

- Theorem: If n^2 is positive, then n is positive.
  - "Proof:" Suppose n^2 is positive. If n is positive, n^2 is positive. Therefore *n* is positive.
  - Why: Let P(n) be "n is positive" and Q(n) be "n^2 is positive". $\forall n(P(n) \rightarrow Q(n))$, Q(n) doesn't mean P(n)
- How to choose right method?
  - Black magic…
- Just a beginning
  - We'll keep things simple in the course – I'll allow lots of leeway.

11 Sets

- A **set** is an unordered collection of objects.
- Useful way of grouping discrete structures together.
- Everything builds on top of this abstract concept.
- The objects in a set are also called the **elements** or **members** of a set.
  - Notation $\in$
  - Duplicates make no difference, i.e., {1, 3, 5} = {1, 1, 3, 3, 5, 5}
- How to describe?
  - List all members V = {a, e, i, o, u}
  - Set of integers less than 100 = {1, 2, 3, …, 99}

12 Common sets

- **N** = natural numbers = {0, 1, 2, 3, …} – sometimes not zero
- **Z** = integers = {…, -2, -1, 0, 1, 2, …}
- $\mathbf{Z}^+$ = positive integers = {1, 2, …}
- **Q** = rational numbers = $\{p/q \mid p \in \mathbf{Z}, q \in \mathbf{Z}, q \neq 0\}$
- **R** = real numbers (incl. irrationsal)

13 Other notations

- Set builder: State the property they must have to be members

- O = {x | x is an odd positive integer less than 10}
- Venn diagram
  - Remember Universal Set U is the contents of the box
  - Venn diagram showing vowels?
- Empty set: { } or ∅
- **Equal:** Two sets are equal iff they have the same elements.
- **Subset:** A ⊆ B -- A is a *subset* of B if and only if every element of A is also an element of B.
  - For any set S, ∅ ⊆ S and S ⊆ S

14 More set notation

- Proper subset, ⊂
- Show equality by showing each set is a subset of the other (can't be proper)
- Can nest sets within sets
- Cardinality of a set is |S|, number of *distinct* elements in a set, assuming S is *finite*.
- **Power set** of S is the set of all subsets of S, or P(S).
  - P({0, 1, 2}) = {∅, {0}, {1}, …, {0, 1, 2}}
  - P(∅) = {∅}
  - Power set of a set has 2^n elements.

15 Tuples and Cartesian Product

- Generally ordered, as opposed to sets
  - Ordered n-tuple (a_1, …, a_n)
  - Cartesian product of set A and set B, A x B, is the set of all ordered pairs (a,b) where a ∈ A and b ∈ B, i.e.,
  - A x B = {(a,b) | a ∈ A ∧ b ∈ B}
  - Example: A is the set of students, and B the set of courses at a university – Cartesian product is the set of all possible enrollments of students in courses.
- N-way Cartesian product generates n-tuples (not nested tuples!)

16 Set notation with quantifiers

- As I showed last time…
- ∀x∈**R** (x^2 ≥ 0)
- Can also use set builder notation

17 Set operations

- Union (∪) is the set that contains those elements that are in A, B, or both.
  - Generally don't include duplicates
- Intersection (∩) is the set containing elements in *both* A and B
- Illustrate using Venn diagrams
- **Disjoint** if intersection is the empty set.
- **Difference**, or A-B, is the set containing elements in A but not in B.
- **Complement,** or A with a bar on top, is the complement of A with respect to U (the universal set).
  - Difference is the intersection of A and the complement of U

18 Set identities

- Page 89, similar to logical equivalences
- Can use direct proof or *membership table* to demonstrate
  - Example: prove that A ∩ (B ∪ C) = (A ∩ B) ∪ (A ∩ C)

19 Generalized union/intersection, computer representation

- Concept remains same; notation is slightly different
  - Page 92/93
- How to represent in a computer?

– If finite, use bitstrings, assuming ordered.
– Can use NOT, AND, OR to do complement, intersection, and union.

20 Functions

- A *function f* from (set) A to (set) B is an assignment of exactly one element of B to each element of A.
  - f(a) = b if b is the unique element of B assigned by the function f to the element a of A.
    - b is the *image* of A and a is a *preimage* of b.
    - *range* of f is the set of all *images* of elements of A.
  - If f is a function from A to B, we can write f: A → B.
    - A is the *domain* of f and B is the *codomain* of f.
    - f "maps" A to B.
- Examples
  - Page 97 for a visual representation
  - f: **Z** → **Z** assigns the square of an integer to this integer. Then, $f(x) = x^2$.
    - Note range and codomain may not be the same.

21 Function Operations

- Real-valued functions can be added and/or multiplied – just "combine" the individual functions
  - If $f_1(x) = x^2$ and $f_2(x) = x - x^2$, $(f_1 + f_2)(x) = x$ and $(f_1f_2)(x) = x^3 - x^4$.
- If a subset of a domain is defined, you can define its image as well.
  - $f(S) = \{f(s) \mid s \in S\}$.

22 One-to-one vs. onto

- Functions always map each preimage to a unique value.
- One-to-one suggests that every mapping maps to a unique image, i.e., f(x) = f(y) implies that x = y.
  - *"Injection*"
  - $f(x) = x^2$ is *not* one-to-one, because of negative values.
  - f(x) = x+1 is one-to-one.
- Onto suggests that each element of the codomain has a preimage.
  - $f(x) = x^2$ is *not* onto, because of negative or skipped integers
  - f(x) = x+1 is onto (infinite trick)
  - *"Surjection"*
- One-to-one correspondence/bijection if it's both.
- See diagram on page 101.

23 Inverse and composition

- The inverse of a function, f^{-1}, assigns to an element *b* in B the unique element *a* in A such that f(a) = b.
  - Must be one-to-one correspondence (i.e, one-to-one and onto).
- The composition (f o g)(a) = f(g(a))
  - *Not* the same as (g o f)(a).

24 Graphs, miscellaneous functions

- Exactly what you'd expect…
  - Although not necessarily continuous
- Floor (or greatest integer) function ($\lfloor x \rfloor$) returns the largest integer that is less than or equal to a real number x.
- Ceiling function ($\lceil x \rceil$) returns the smallest integer that is greater than or equal to a real number x.
- Graphs of both on page 106
  - Note open circles mean open intervals, e.g., floor has same value from [n, n+1) and ceiling has the same value from (n, n+1]
- Various useful properties on page 107
  - Is $\lceil x+y \rceil = \lceil x \rceil + \lceil y \rceil$?

25 Next time

- Algorithms, growth
- Integers and integer algorithms