CSE 20
DISCRETE MATH

Fall 2017

http://cseweb.ucsd.edu/classes/fa17/cse20-ab/

Final exam

HW 8 due Sat Dec 9

The final exam is **Saturday December 16 11:30am-2:29pm**.

Lecture A will take the exam in CENTR 115

Lecture B will take the exam in CENTR 119

Review session **TBA**

office hour schedule on Google calendar

Complete **post-quarter survey** for credit.

Complete **HW review form** to be eligible to drop a HW.

Review quizzes from all weeks still open for studying.

1. Algorithms
2. Number systems and integer operations
3. Propositional Logic
4. Predicates & Quantifiers
5. Proof strategies
6. Sets
7. Induction & Recursion
8. Functions & Cardinalities of sets
9. Binary relations & Modular arithmetic

Algorithms

- Trace pseudocode given input.
- Explain the higher-level function of an algorithm expressed with pseudocode.
- Identify and explain (informally) whether and why an algorithm expressed in pseudocode terminates for all input.
- Describe and use classical algorithms:
  - Addition and multiplication of integers expressed in some base
- Define the greedy approach for an optimization problem.
- Write pseudocode to implement the greedy approach for a given optimization problem.

Pseudocode

Review pseudocode from midterm exam

```
procedure notSoGreedyChange(n : a positive integer)
    for i := 1 to r
        d_i := 0
    while n > 0
        for i := 1 to r
            if n ≥ c_i
                d_i := d_i + 1
                n := n − c_i
```

Nested? Incrementing by what in for loops?

Coin-changing

For which values can you make change using just 2c and 5c coins?

7¢, all pos ints except 1, 3

Claim: For each pos int except 1, 3 that value can be expressed as sum of nonneg multiples of 2, 5.

Pf By Cases. Let n be pos int (not 1, 3)

Case ① n even So by def, there's pos k where n=2k. So use k 2¢ coins.

Number systems and integer representations

- Convert between positive integers written in any base b, where b >1.
- Define the decimal, binary, hexadecimal, and octal expansions of a positive integer.
- Describe and use algorithms for integer operations based on their expansions
- Relate algorithms for integer operations to bitwise boolean operations.
- Correctly use XOR and bit shifts.
- Define and use the DIV and MOD operators.

Arithmetic + Representations

Rosen p. 251

Convert $(2A)_{16}$ to …

A. binary (base ?)
B. decimal (base ?)
C. octal (base ?)
D. ternary (base ?)
E. All of the above

Hexadecimal digits

0	8
1	9
2	A
3	B
4	C
5	D
6	E
7	F

Propositional Logic

- Describe the uses of logical connectives in formalizing natural language statements, bit operations, guiding proofs and rules of inference.
- Translate sentences from English to propositional logic using appropriate propositional variables and boolean operators.
- List the truth tables and meanings for negation, conjunction, disjunction, exclusive or, conditional, biconditional operators.
- Evaluate the truth value of a compound proposition given truth values of its constituent variables.
- Form the converse, contrapositive, and inverse of a given conditional statement.
- Relate boolean operations to applications: Complex searches, Logic puzzles, Set operations and spreadsheet queries, Combinatorial circuits
- Prove propositional equivalences using truth tables
- Prove propositional equivalences using other known equivalences, e.g. DeMorgan’s laws, Double negation laws, Distributive laws, etc.
- Identify when and prove that a statement is a tautology or contradiction
- Identify when and prove that a statement is satisfiable or unsatisfiable, and when a set of statements is consistent or inconsistent.
- Compute the CNF and DNF of a given compound proposition.

Conditionals

Rosen p. 6-10

Which of these compound propositions **is** logically equivalent to

$$\neg((p \to \neg q) \to r)$$

A. $(p \to \neg q) \to \neg r$

B. $\neg r \to \neg(p \to \neg q)$

C. $(q \vee r) \to (\neg p \wedge \neg r)$

D. $\neg(p \to \neg q) \vee r$

E. None of the above.

p	q	p → q
T	T	T
T	F	F
F	T	T
F	F	T

Conditionals

Rosen p. 6-10

Which of these compound propositions **is** logically equivalent to

$$\neg((p \to \neg q) \to r)$$

A. $(p \to \neg q) \to \neg r$

B. $\neg r \to \neg(p \to \neg q)$

C. $(q \vee r) \to (\neg p \wedge \neg r)$

D. $\neg(p \to \neg q) \vee r$

E. None of the above.

Normal forms:

A. Do you want to find equivalent CNF and DNF?
B. Just find DNF?
C. Just find CNF?
D. Neither.

Predicates & Quantifiers

- Determine the truth value of predicates for specific values of their arguments
- Define the universal and existential quantifiers
- Translate sentences from English to predicate logic using appropriate predicates and quantifiers
- Use appropriate Boolean operators to restrict the domain of a quantified statement
- Negate quantified expressions
- Translate quantified statements to English, even in the presence of nested quantifiers
- Evaluate the truth value of a quantified statement with nested quantifiers

Evaluating quantified statements *Rosen p. 64 #1*

$$\forall x \forall y (x < y \rightarrow \exists z (x < z < y))$$

In which domain(s) is this statement true?

A. All positive real numbers.
B. All positive integers.
C. All real numbers in closed interval [0,1].
D. The integers 1,2,3.
E. The power set of {1,2,3}

meaningless !

Proof strategies

- Distinguish between a theorem, an axiom, lemma, a corollary, and a conjecture.
- Recognize direct proofs
- Recognize proofs by contraposition
- Recognize proofs by contradiction
- Recognize fallacious “proofs”
- Evaluate which proof technique(s) is appropriate for a given proposition: Direct proof, Proofs by contraposition, Proofs by contradiction, Proof by cases, Constructive existence proofs, induction
- Correctly prove statements using appropriate style conventions, guiding text, notation, and terminology

A sample proof by contradiction

- Theorem: There are infinitely many prime numbers.

pos int that doesn't have any prime divisors (other than itself)

pos int whose only factors are 1 and itself (and >1)

Proof: Assume there are finitely many prime numbers. Let n ∈ ℕ be the # of primes and list all primes:

$p_1, \ldots, p_n$ (>1)

WTS this leads to a contradiction

Goal: Find a prime number not in list

★ $p_1 \cdots p_n + 1 > p_i$ so not $\neq p_i$

and p_i is not a factor of ★. ★ is prime not in list!

Sets

- Define and differentiate between important sets: **N, Z, Z+, Q, R, R+, C**, empty set, {0,1}*
- Use correct notation when describing sets: {...}, intervals, set builder
- Define and prove properties of: subset relation, power set, Cartesian products of sets, union of sets, intersection of sets, disjoint sets, set differences, complement of a set
- Describe computer representation of sets with bitstrings

Power set example

Power set: For a set S, its power set is the set of all subsets of S.

$$\mathcal{P}(S) = \{A \mid A \subseteq S\}$$

Which of the following is **not** true (in general)?

A. $S \in \mathcal{P}(S)$
B. $\emptyset \in \mathcal{P}(S)$
C. $S \subseteq \mathcal{P}(S)$
D. $\emptyset \in S$
E. None of the above

Power set example

Power set: For a set S, its power set is the set of all subsets of S.

$$\mathcal{P}(S) = \{A \mid A \subseteq S\}$$

A= { 1,2,3 }

Give an example of a (well-defined) one-to-one function from the set A to its power set (or explain why this is impossible).

Give an example of a (well-defined) onto function from the set A to its power set (or explain why this is impossible).

Give an example of a (well-defined) function from the set A to its power set that is neither one-to-one nor onto (or explain why this is impossible).

Induction and recursion

- Explain the steps in a proof by mathematical induction
- Explain the steps in a proof by strong mathematical induction
- Use (strong) mathematical induction to prove correctness of identities and inequalities
- Use (strong) mathematical induction to prove properties of algorithms
- Use (strong) mathematical induction to prove properties of geometric constructions
- Apply recursive definitions of sets to determine membership in the set
- Use structural induction to prove properties of recursively defined sets

Structural induction

Theorem: For any bit string w, zeros(w) ≤ l(w).

A. What does this mean? How to prove it?
B. Just talk about what it means.
C. How does structural induction apply?
D. Neither.

Functions & Cardinality of sets

- Represent functions in multiple ways
- Define and prove properties of domain of a function, image of a function, composition of functions
- Determine and prove whether a function is one-to-one
- Determine and prove whether a function is onto
- Determine and prove whether a function is bijective
- Apply the definition and properties of floor function, ceiling function, factorial function
- Define and compute the cardinality of a set
  - Finite sets
  - countable sets
  - uncountable sets
- Use functions to compare the sizes of sets
- Use functions to define sequences: arithmetic progressions. geometric progressions
- Use and prove properties of recursively defined functions and recurrence relations (using induction)
- Use and interpret Sigma notation

Cardinality and subsets

Suppose A and B are sets and $A \subseteq B$.

A. If A is infinite then B is finite.
B. If A is countable then B is countable. Could fix A→B.
C. If B is infinite then A is finite.
D. If B is uncountable then A is uncountable.
E. None of the above.

Binary relations

- Determine and prove whether a given binary relation is
  - symmetric
  - reflexive
  - transitive
- Represent equivalence relations as partitions and vice versa
- Define and use the congruence modulo m equivalence relation

Properties of binary relations

Over the set $\mathbf{Z}^+$

A. Define a binary relation that is reflexive, not symmetric, and not transitive.
B. Define a binary relation that is not reflexive, but is symmetric and transitive.
C. Define an equivalence relation with exactly three distinct equivalence classes.
D. Define an equivalence relation with infinitely many distinct equivalence classes, each of finite size.

Modular arithmetic

Rosen p. 288

Solve the congruences

$$x \equiv 3 \pmod{4} \qquad \text{i.e.} \qquad x \mathbf{\ mod\ } 4 = 3$$

$$5 + x \equiv 3 \pmod{7} \qquad \text{i.e.} \qquad 5 + x \mathbf{\ mod\ } 7 = 3$$

$$5x \equiv 3 \pmod{7} \qquad \text{i.e.} \qquad 5x \mathbf{\ mod\ } 7 = 3$$

Reminders

1. Algorithms
2. Number systems and integer operations
3. Propositional Logic
4. Predicates & Quantifiers
5. Proof strategies
6. Sets
7. Induction & Recursion
8. Functions & Cardinalities of sets
9. Binary relations and

Final exam Saturday December 16 11:30am in announced rooms.
See all details on website and Piazza.