Discrete Math Review
(Rosen, Chapter 1.1 – 1.7, 5.5)

TOPICS
- Sets and Functions
- Propositional and Predicate Logic
- Logical Operators and Truth Tables
- Logical Equivalences and Inference Rules
- Proof Techniques
- Program Correctness

Discrete Math Review

- What you should know about discrete math before the next midterm!
- Less theory, more problem solving, will be repeated in recitation and homework.

Set Definitions

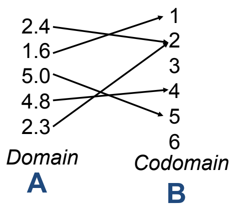
- An unordered collection of objects (elements)
- Membership: $1 \in \{1, 2, 3, 4, 5\}$, $6 \notin \{1, 2, 3\}$
- Builder: $O = \{ x \in N \mid x \text{ is odd and } x < 10\}$
- Equality: A = B if exactly the same elements
- Union: $A \cup B$, set of elements in A or B
- Intersection: $A \cap B$, set of elements in A and B
- Difference: A – B, set of elements in A but not B
- Complement: $\overline{A}$, set of elements not in A but in U

Set Definitions, continued

- Empty Set: $\varnothing$, or { }, subset of all sets
- Cardinality: V = {a, e, i, o, u}, so |V| = 5
- Subset: $A \subseteq B$, all elements in A are in B
- Proper Subset: $A \subset B$, same as subset but $A \neq B$
- Power Set: P(A), set of all subsets, $|P(A)| = 2^{|A|}$
- Tuples: order matters, duplicates ok, (1, 3, 2)
- Cartesian Product: A x B, |A x B| = |A| x |B|
- Identities: $A \cup \{\} = A$, $A \cap \{\} = \{\}$

Function Definitions

- A function **f** from sets **A** to **B** assigns exactly one element of **B** to each element of **A**.
- Example: the **floor** function



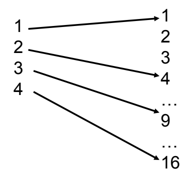
What's the difference between codomain and range?

Range: {1,2,4,5}

Range contains the codomain values that A maps to

1 to 1 Functions

- A function **f** is said to be *one-to-one* or *injective* if and only if **f(a) = f(b)** implies that **a = b** for all **a** and **b** in the domain of **f**.
- Example: the **square** function from $\mathbf{Z}^+$ to $\mathbf{Z}^+$



Increasing Functions

- A function **f** whose domain and co-domain are subsets of the set of real numbers is called *increasing* if **f(x) <= f(y)** and *strictly increasing* if **f(x) < f(y)**
- Is **floor** an example?

1.5 < 1.7 and floor(1.5) = 1 = floor(1.7)
1.2 < 2.2 and floor(1.2) = 1 < 2 = floor(2.2),

So YES floor is increasing

BUT it is **NOT** STRICTLY increasing

- Is **square** an example?

When mapping Z to Z or R to R:
square(-2) = 4 > 1 = square (1) yet -2 < 1

NO square is NOT increasing UNLESS….

Domain is restricted to positive #'s

Sets and Functions

- (1) You should know set and tuple definitions and operations and be able to compute them.
- (2) You should understand sets well enough to determine the truth values of identities.
- (3) You should understand the definitions of the domain, co-domain, and range.
- (4) You should understand functions well enough to determine injective and increasing.

Propositional Logic

- A *proposition* is a statement that is either true or false
- Examples:
  - Fort Collins is in Nebraska (false)
  - Java is case sensitive (true)
  - We are not alone in the universe (?)
- Every proposition is true or false, but its *truth value* may be unknown

Logical Operators

- ¬ logical not (negation)
- ∨ logical or (disjunction)
- ∧ logical and (conjunction)
- ⊕ logical exclusive or
- → logical implication (conditional)
- ↔ logical bi-implication (biconditional)

Truth Tables

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

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

- (5) You should be able to write out the truth table for all logical operators, from memory.

Compound Propositions

- Propositions and operators can be combined into compound propositions.
- (6) You should be able to make a truth table for any compound proposition:

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

English to Propositional Logic

- (7) You should be able to translate natural language to logic (can be ambiguous!):
- English:
  - "If the car is out of gas, then it will stop"
- Logic:
  - p equals "the car is out of gas"
  - q equals "the car will stop"
  - $p \rightarrow q$

Propositional Logic to English

- (8) You should be able to translate propositional logic to natural language:
- Logic:
  - p equals "it is raining"
  - q equals "the grass will be wet"
  - $p \rightarrow q$
- English:
  - "If it is raining, the grass will be wet."

Logical Equivalences: Definition

- Certain propositions are equivalent (meaning they share exactly the same truth values):
- For example:
  - $\neg(p \wedge q) \equiv \neg p \vee \neg q$ De Morgan's
  - $(p \wedge T) \equiv p$ Identity Law
  - $(p \wedge \neg p) \equiv F$ Negation Law

Logical Equivalences: Truth Tables

- (9) And you should know how to prove logical equivalence with a truth table
- For example: $\neg(p \wedge q) \equiv \neg p \vee \neg q$

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

Logical Equivalences: Review

- (10) You should understand the logical equivalences and laws on the course web site.
- You should be able to prove any of them using a truth table that compares the truth values of both sides of the equivalence.
- Memorization of the logical equivalences is not required in this class.

Logical Equivalences (Rosen)

Logical Equivalences

Idempotent Laws
$p \vee p \equiv p$
$p \wedge p \equiv p$

DeMorgan's Laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$
$\neg(p \vee q) \equiv \neg p \wedge \neg q$

Distributive Laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$

Double Negation
$\neg(\neg p) \equiv p$

Absorption Laws
$p \vee (p \wedge q) \equiv p$
$p \wedge (p \vee q) \equiv p$

Associative Laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$
$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$

Commutative Laws
$p \vee q \equiv q \vee p$
$p \wedge q \equiv q \wedge p$

Implication Laws
$p \rightarrow q \equiv \neg p \vee q$
$p \rightarrow q \equiv \neg q \rightarrow \neg p$

Biconditional Laws
$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$
$p \leftrightarrow q \equiv \neg q \leftrightarrow \neg p$

Transformation via Logical Equivalences

(11) You should be able to transform propositions using logical equivalences.

Prove: $\neg p \vee (p \wedge q) \equiv \neg(p \wedge \neg q)$

$\neg p \vee (p \wedge q) \equiv (\neg p \vee p) \wedge (\neg p \vee q)$ — Distributive law
$\equiv T \wedge (\neg p \vee q)$ — Negation law
$\equiv (\neg p \vee q)$ — Domination law
$\equiv \neg(p \wedge \neg q)$ — De Morgan's Law

Vocabulary

- (12) You should memorize the following vocabulary:
  - A *tautology* is a compound proposition that is always true.
  - A *contradiction* is a compound proposition that is always false.
  - A *contingency* is neither a tautology nor a contradiction.
- And know how to decide the category for a compound proposition.

Examples

p	$\neg p$	$p \vee \neg p$	$p \wedge \neg p$
T	F	T	F
F	T	T	F

Result is always true, no matter what A is

Therefore, it is a ***tautology***

Result is always false, no matter what A is

Therefore, it is a ***contradiction***

Logical Proof

- Given a set of *axioms*
  - Statements asserted to be true
- Prove a *conclusion*
  - Another propositional statement
- In other words:
  - Show that the conclusion is true …
  - … whenever the axioms are true

Logical Proof

- (13) You should be able to perform a logical proof via truth tables.
- (14) You should be able to perform a logical proof via inference rules.
- Both methods are described in the following slides.

Method 1: Proof by Truth Table

- Prove that $p \rightarrow q$, given $\neg p$

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

For all rows in which axiom is true, conclusion is true

Thus the conclusion follows from axiom

Method 2: Proof using Rules of Inference

- A *rule of inference* is a proven relation: when the left hand side (LHS) is true, the right hand side (RHS) is also true.
- Therefore, if we can match an axiom to the LHS by substituting propositions, we can assert the (substituted) RHS

Applying rules of inference

- Example rule: $p, p \rightarrow q \therefore q$
  - Read as "*p* and $p \rightarrow q$, therefore *q*"
  - This rule has a name: *modus ponens*
- If you have axioms $r, r \rightarrow s$
  - Substitute *r* for *p*, *s* for *q*
  - Apply modus ponens
  - Conclude *s*

Modus Ponens

- If p, and p implies q, then q

Example:

p = it is sunny, q = it is hot

$p \rightarrow q$, it is hot whenever it is sunny

"Given the above, if it is sunny, it must be hot".

Modus Tollens

- If not q and p implies q, then not p

Example:

p = it is sunny, q = it is hot

$p \rightarrow q$, it is hot whenever it is sunny

"Given the above, if it is not hot, it cannot be sunny."

Rules of Inference (Rosen)

Rules of Inference

Modus Ponens

p
$p \rightarrow q$
———
q

Modus Tollens

$\neg q$
$p \rightarrow q$
———
$\neg p$

Hypothetical Syllogism

$p \rightarrow q$
$q \rightarrow r$
———
$p \rightarrow r$

Addition

p
———
$p \vee q$

Resolution

$p \vee q$
$\neg p \vee r$
———
$q \vee r$

Disjunctive Syllogism

$p \vee q$
$\neg p$
———
q

Simplification

$p \wedge q$
———
p

Conjunction

p
q
———
$p \wedge q$

A Simple Proof: Problem Statement

Example of a complete proof using inference rules, from English to propositional logic and back:

- If you don't go to the store, then you cannot not cook dinner. (axiom)
- If you cannot cook dinner or go out, you will be hungry tonight. (axiom)
- You are not hungry tonight, and you didn't go to the store. (axiom)
- You must have gone out to dinner. (conclusion)

A Simple Proof: Logic Translation

- p: you go to the store
- q: you can cook dinner
- r: you will go out
- s: you will be hungry
- AXIOMS: $\neg p \rightarrow \neg q$, $\neg(q \vee r) \rightarrow s$, $\neg s$, $\neg p$
- CONCLUSION: r

A Simple Proof: Applying Inference

1.	$\neg p \rightarrow \neg q$	Axiom
2.	$\neg(q \vee r) \rightarrow s$	Axiom
3.	$\neg s$	Axiom
4.	$\neg p$	Axiom
5.	$\neg q$	Modus Ponens (1, 4)
6.	$q \vee r$	Modus Tollens (2, 3)
7.	r	Disjunctive Syllogism (5, 6)

CONCLUSION: You must have gone out to dinner!

Predicate Logic

- (15) You should recognize predicate logic symbols, i.e. quantifications.
- Quantification express the extent to which a predicate is true over a set of elements:
  - Universal ∀, “for all”
  - Existential ∃, “there exists”
- (16) You should able to translate between predicate logic and English, in both directions.

Predicate Logic (cont’d)

- Specifies a proposition (and optionally a domain), for example:
  - $\exists x \in N, -10 < x < -5$ // False, since no negative x
  - $\forall x \in N, x > -1$ // True, since no negative x
- (17) Must be able to find examples
  - to prove ∃, e.g. $\exists x \in Z, -1 < x < 1$, $x = 0$
- (18) Must be able to find counterexamples
  - to disprove ∀, e.g. $\forall x \in Z, x > -1$, $x = -2$

Direct Proof (19): Show that ***5x + 3y*** is even when ***x*** and ***y*** are odd integers.

	Step	Reason
1.	**O(x) ∧ O(y) → E(5x + 3y)**	Hypothesis
2.	**O(x) → x = 2j+1, O(y) → y = 2k+1**	Odd Definition
3.	**E(5(2j + 1) + 3(2k + 1))**	Substitution
4.	**E(10j + 5 + 6k + 3)**	Algebra
5.	**E(2(5j + 3k + 4)) = true**	Even Definition
6.	**∴ E(5x + 3y) = true**	Proves hypothesis

Contrapositive Proof (20): Show that when ***5xy*** is even, then ***x*** or ***y*** is even

	Step	Reason
1.	**E(5xy) → E(x) ∨ E(y)**	**Hypothesis**
2.	**¬(E(x) ∨ E(y)) → ¬E(5xy)**	**Contrapositive**
3.	**O(x) ∧ O(y) → O(5xy)**	**De Morgan’s**
4.	**O(x) → x = 2j+1, O(y) → y = 2k+1**	**Odd Definition**
5.	**O(5(2j + 1)(2k + 1))**	**Substitution**
6.	**O(20jk + 10j + 10k + 5)**	**Algebra**
7.	**O(2(10jk + 5j + 5k + 2) + 1) = true**	**Odd Definition**
8.	**∴O(5xy) = true**	**Proves Contrapositive**

Contradiction Proof (21): Show that when ***5xy*** is even, then ***x*** or ***y*** is even

	Step	Reason
1.	E(5xy) → E(x) ∨ E(y)	Hypothesis
2.	E(5xy) ∧ ¬(E(x) ∨ E(y))	Contradiction
3.	E(5xy) ∧ O(x) ∧ O(y))	De Morgan's
4.	O(x) → x = 2j+1, O(y) → y = 2k+1	Odd Definition
5.	E(5(2j + 1)(2k + 1))	Substitution
6.	E(20jk + 10j + 10k + 5)	Algebra
7.	E(2(10jk + 5j + 5k + 2) + 1) = false	Odd Definition!
8.	∴ E(5xy) = false	Disproves Contradiction

Proof by Cases (22): Given two real numbers ***x*** and ***y***, **|*xy*| = |*x*||*y*|**

Case 1: x>=0, y>=0, xy>=0 so |xy|=xy
and |x|=x and |y|=y so |x||y|=xy

Case 2: x<0, y>=0, xy<0 so |xy|=-xy
and |x|=-x and |y|=y so |x||y|=-xy

Case 3: x>=0, y<0, xy<0 so |xy|=-xy
and |x|=x and |y|=-y so |x||y|=-xy

Case 4: x<0, y<0, xy>=0 so |xy|=xy
and |x|=-x and |y|=-y so |x||y|=xy

Pre and Post Conditions (23)

```
public static int foo(int x) {
    // Precondition: -4 <= x <= 3
    return (x * x + 2 * x - 5);
    // Postcondition -6 <= return value <= 10
}
```

f(-4) = 3, f(-3) = -2, f(-2) = -5, f(-1) = -6
f(0) = -5, f(1) = -2, f(2) = 3, f(3) = 10

Pre and Post Conditions (24)

```
public static int foo(int x) {
    // Precondition: -4 <= x <= 2
    return (x * x + 2 * x - 5);
    // Postcondition -6 <= return <= 3
}
```

f(-5) = 10, f(-4) = 3, f(-3) = -2, f(-2) = -5,
f(-1) = -6, f(0) = -5, f(1) = -2, f(2) = 3, f(3) = 10

Loop Invariants (25)

```
int x = 1, y = 2, z = -5;
while (x <= 5) {
   z += y;
   x++;
}
// Loop invariants
y = 2, 1 <= x <= 6, -5 <= z <= 5
```