Discrete Math Review
(Rosen, Chapter 1.1 – 1.6)

TOPICS

- 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 propositional and predicate logic before the next midterm!
- Less theory, more problem solving, will be repeated in recitation and homework.

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

- $\neg$ logical not (negation)
- $\vee$ logical or (disjunction)
- $\wedge$ logical and (conjunction)
- $\oplus$ logical exclusive or
- $\rightarrow$ logical implication (conditional)
- $\leftrightarrow$ 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

- (1) 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.
- (2) 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

- (3) 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

- (4) 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

- (5) 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*	*¬p*	*¬q*	*(p ∧ q)*	*¬(p ∧ q)*	*¬p ∨ ¬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

- (6) 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

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

Prove: $\neg p \lor (p \land q) \equiv \neg(p \land \neg q)$

$$\begin{aligned} \neg p \lor (p \land q) &\equiv (\neg p \lor p) \land (\neg p \lor q) \\ &\equiv T \land (\neg p \lor q) \\ &\equiv (\neg p \lor q) \\ &\equiv \neg(p \land \neg q) \end{aligned}$$

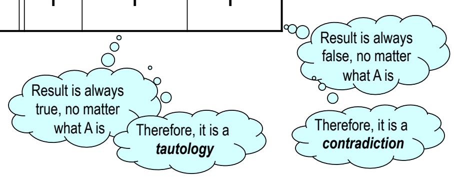
- Distributive law
- Negation law
- Domination law
- De Morgan's Law

Vocabulary

- (8) 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	$\neg p$	$p \lor \neg p$	$p \land \neg p$
T	F	T	F
F	T	T	F



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

- (9) You should be able to perform a logical proof via truth tables.
- (10) 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

$$\frac{p \qquad p \rightarrow q}{q}$$

Modus Tollens

$$\frac{\neg q \qquad p \rightarrow q}{\neg p}$$

Hypothetical Syllogism

$$\frac{p \rightarrow q \qquad q \rightarrow r}{p \rightarrow r}$$

Addition

$$\frac{p}{p \vee q}$$

Resolution

$$\frac{p \vee q \qquad \neg p \vee r}{q \vee r}$$

Disjunctive Syllogism

$$\frac{p \vee q \qquad \neg p}{q}$$

Simplification

$$\frac{p \wedge q}{p}$$

Conjunction

$$\frac{p \qquad q}{p \wedge q}$$

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

- (11) 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 $\forall$, “for all”
  - Existential $\exists$, “there exists”
- (12) 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
- (13) Must be able to find examples
  - to prove $\exists$, e.g. $\exists x \in Z, -1 < x < 1$, $x = 0$
- (14) Must be able to find counterexamples
  - to disprove $\forall$, e.g. $\forall x \in Z, x > -1$, $x = -2$

Direct Proof (15): 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 (16): 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 (17): 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 (18): 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 (19)

```
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 (20)

```
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 (21)

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