Discrete Mathematics (2009 Spring) Foundations of Logic (§1.1-§1.4, 4 hours)

Chih-Wei Yi

Dept. of Computer Science
National Chiao Tung University

March 1, 2010

Foundations of Logic

- Mathematical Logic is a tool for working with complicated compound statements. It includes:
    - A language for expressing them.
    - A concise notation for writing them.
    - A methodology for objectively reasoning about their truth or falsity.
    - It is the foundation for expressing formal proofs in all branches of mathematics.

Overview

- Propositional logic (§1.1-§1.2):
  - Basic definitions (§1.1)
  - Equivalence rules & derivations (§1.2)
- Predicate logic (§1.3-§1.4)
  - Predicates
  - Quantified predicate expressions
  - Equivalences & derivations

§1.1 Propositional Logic

Definition of a Proposition

Definition

A proposition ($p, q, r, \ldots$) is a declarative sentence with a definite meaning, having a truth value that's either true (T) or false (F) (**never** both, neither, or somewhere in between).

- However, you might not know the actual truth value, and it might be situation-dependent.

Examples of Propositions

Example

The following statements are propositions:

- "It is raining." (In a given situation.)
- "Washington D.C. is the capital of China."
- "$1 + 2 = 3$"

But, the followings are **NOT** propositions:

- "Who's there?" (interrogative, question)
- "La la la la la." (meaningless interjection)
- "Just do it!" (imperative, command)
- "$1 + 2$" (expression with a non-true/false value)

Propositional Logic

- Propositional Logic is the logic of compound statements built from simpler statements using so-called Boolean connectives.
- Some applications in computer science:
  - Design of digital electronic circuits.
  - Expressing conditions in programs.
  - Queries to databases & search engines.

Operators / Connectives

- An operator or connective combines one or more operand expressions into a larger expression (e.g., "$+$" in numeric exprs).
- Unary operators take 1 operand (e.g., -3); binary operators take 2 operands (e.g., 3×4).
- Propositional or Boolean operators operate on propositions or truth values instead of on numbers.

Some Popular Boolean Operators

Formal Name	Nickname	Arity	Symbol
Negation operator	NOT	Unary	$\neg$
Conjunction operator	AND	Binary	$\wedge$
Disjunction operator	OR	Binary	$\vee$
Exclusive-OR operator	XOR	Binary	$\oplus$
Implication operator	IMPLIES	Binary	$\longrightarrow$
Biconditional operator	IFF	Binary	$\longleftrightarrow$

The Negation Operator

- The unary negation operator "¬" (NOT) transforms a prop. into its logical negation.
  - For example, if p = "I have brown hair.", then $\neg p$ = "I do **not** have brown hair.".
- Truth table for NOT:

Operand column

Result column

p	$\neg p$
T	F
F	T

T :≡True; F :≡False
":≡" means "is defined as"

The Conjunction Operator

- The binary conjunction operator "$\wedge$" (AND) combines two propositions to form their logical conjunction.

Example

$p =$ "I will have salad for lunch."
$q =$ "I will have steak for dinner."
$p \wedge q =$ "I will have salad for lunch **and** I will have steak for dinner."

Remember: "$\wedge$" points up like an "A", and it means "AND"

Conjunction Truth Table

- Note that a conjunction $p_1 \wedge p_2 \wedge \cdots \wedge p_n$ of n propositions will have 2^n rows in its truth table.
- $\neg$ and $\wedge$ operations together are sufficient to express any Boolean truth table with only 1 True value.

Operand columns

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

The Disjunction Operator

- The binary disjunction operator "$\vee$" (OR) combines two propositions to form their logical disjunction.

Example

$p =$ "My car has a bad engine."
$q =$ "My car has a bad carburetor."
$p \vee q =$ "Either my car has a bad engine, **or** my car has a bad carburetor."

Meaning is like "and/or" in English.

Disjunction Truth Table

- Note that $p \vee q$ means that p is true, or q is true,**or both** are true!
- So, this operation is also called inclusive or, because it **includes** the possibility that both p and q are true.
- $\neg$ and $\vee$ operations together are sufficient to express any Boolean truth table with only 1 False value.

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

Note difference from AND

Precedence of Logical Operators

- Use parentheses to group sub-expressions, for example "I just saw my old *f*riend, and either he's *g*rown or I've *s*hrunk." $= f \wedge (g \vee s)$.
  - $(f \wedge g) \vee s$ would mean something different.
  - $f \wedge g \vee s$ would be ambiguous.
- By convention, "$\neg$" takes precedence over both "$\wedge$" and "$\vee$".
  - $\neg s \wedge f$ means $(\neg s) \wedge f$, **not** $\neg(s \wedge f)$.

A Simple Exercise

- Let $p =$ "It rained last night.", $q =$ "The sprinklers came on last night.", and $r =$ "The lawn was wet this morning."
- Translate each of the following into English:
  - $\neg p$
    - "It didn't rain lastnight."
  - $r \wedge \neg p$
  - $\neg r \vee p \vee q$

A Simple Exercise

- Let $p =$ "It rained last night.", $q =$ "The sprinklers came on last night.", and $r =$ "The lawn was wet this morning."
- Translate each of the following into English:
  - $\neg p$
    - "It didn't rain lastnight."
  - $r \wedge \neg p$
    - "The lawn was wet this morning, and it didn't rain last night."
  - $\neg r \vee p \vee q$

A Simple Exercise

- Let $p =$ "It rained last night.", $q =$ "The sprinklers came on last night.", and $r =$ "The lawn was wet this morning."
- Translate each of the following into English:
  - $\neg p$
    - "It didn't rain lastnight."
  - $r \wedge \neg p$
    - "The lawn was wet this morning, and it didn't rain last night."
  - $\neg r \vee p \vee q$
    - "Either the lawn wasn't wet this morning,or it rained lastnight, or thes prinklers came on lastnight."

The Exclusive OR Operator

- The binary exclusive-or operator "$\oplus$" (XOR) combines two propositions to form their logical "exclusive or" (exjunction?).
  - $p =$ "I will earn an A in this course,"
  - $q =$ "I will drop this course,"
  - $p \oplus q =$ "I will either earn an A for this course, or I will drop it (but not both!)"

Exclusive-OR Truth Table

- Note that $p \oplus q$ means that p is true, or q is true, but **not both**!
- This operation is called exclusive or, because it **excludes** the possibility that both p and q are true.
- “¬“ and “⊕” together are **not** universal.

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

Note difference from OR.

Natural Language is Ambiguous

- Note that English "or" can be ambiguous regarding the "**both**" case!
  - "Pat is a singer or Pat is a writer." - $\vee$
  - "Pat is a man or Pat is a woman." - $\oplus$
- Need context to disambiguate the meaning!
- For this class, assume "or" means inclusive.

p	q	p "or" q
F	F	F
F	T	T
T	F	T
T	T	?

The Implication Operator

- The implication statement "p implies q" is denoted by

$$\overbrace{p}^{\text{antecedent}} \rightarrow \overbrace{q}^{\text{consequent}}$$

- In other words, if p is true, then q is true; but if p is not true, then q could be either true or false.

Example

Let $p =$ "You study hard." and $q =$ "You will get a good grade.". Then, $p \rightarrow q =$ "If you study hard, then you will get a good grade." (else, it could go either way)

Implication Truth Table

- $p \rightarrow q$ is **false** only when p is true but q is **not** true.
- $p \rightarrow q$ does **not** say that p causes q!
- $p \rightarrow q$ does **not** require that p or q are ever true! e.g. "(1=0)→pigs can fly" is TRUE!

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

The only False case!

Examples of Implications

- "If this lecture ends, then the sun will rise tomorrow." True or False?
  - True
- "If Tuesday is a day of the week, then I am a penguin." True or False?
- "If 1+1=6, then Bush is president." True or False?
- "If the moon is made of green cheese, then I am richer than Bill Gates." True or False?

Examples of Implications

- "If this lecture ends, then the sun will rise tomorrow." True or False?
  - True
- "If Tuesday is a day of the week, then I am a penguin." True or False?
  - False
- "If 1+1=6, then Bush is president." True or False?
- "If the moon is made of green cheese, then I am richer than Bill Gates." True or False?

Examples of Implications

- "If this lecture ends, then the sun will rise tomorrow." True or False?
  - True
- "If Tuesday is a day of the week, then I am a penguin." True or False?
  - False
- "If 1+1=6, then Bush is president." True or False?
  - True
- "If the moon is made of green cheese, then I am richer than Bill Gates." True or False?

Examples of Implications

- "If this lecture ends, then the sun will rise tomorrow." True or False?
  - True
- "If Tuesday is a day of the week, then I am a penguin." True or False?
  - False
- "If 1+1=6, then Bush is president." True or False?
  - True
- "If the moon is made of green cheese, then I am richer than Bill Gates." True or False?
  - True

Why Does This Seem Wrong?

- Consider a sentence like,
  - "If I wear a red shirt tomorrow, then the U.S. will attack Iraq the same day."
- In logic, we consider the sentence **True** so long as either I don't wear a red shirt, or the US attacks.
- But in normal English conversation, if I were to make this claim, you would think I was lying.
  - Why this discrepancy between logic & language?

Resolving the Discrepancy

- In English, a sentence "if p then q" usually really *implicitly* means something like,
  - "*In all possible situations*, if p then q."
    - That is, "For p to be true and q false is *impossible*."
    - Or, "I *guarantee* that no matter what, if p, then q."
- This can be expressed in *predicate logic* as:
  - "For all situations s, if p is true in situation s, then q is also true in situation s"
  - Formally, we could write: $\forall s, P(s) \to Q(s)$.
- This sentence is logically **False** in our example, because for me to wear a red shirt and the U.S. not to attack Iraq is a possible (even if not actual) situation.
  - Natural language and logic then agree with each other.

English Phrases Meaning p -> q

- "p implies q"
- "if p ,then q"
- "if p, q"
- "when p, q"
- "whenever p, q"
- "q if p"
- "q when p"
- "q whenever p"
- "p only if q"
- "p is sufficient for q"
- "q is necessary for p"
- "q follows from p"
- "q is implied by p"

- We will see some equivalent logic expressions later.

Converse, Inverse, Contrapositive

- Some terminology, for an implication $p \longrightarrow q$:
  - Its converse is $q \longrightarrow p$.
  - Its inverse is $\neg p \longrightarrow \neg q$.
  - Its contrapositive is $\neg q \longrightarrow \neg p$.
- One of these three has the same meaning (same truth table) as $p \longrightarrow q$. Can you figure out which?

How Do We Know For Sure?

- Proving the equivalence of $p \rightarrow q$ and its contrapositive using truth tables:

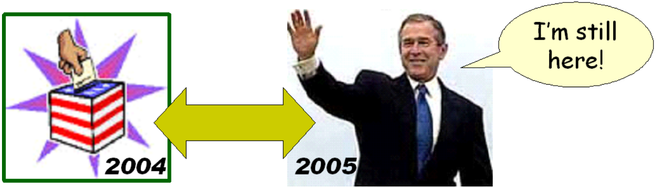
p	q	$\neg q$	$\neg p$	$p \rightarrow q$	$\neg q \rightarrow \neg p$
F	F	T	T	T	T
F	T	F	T	T	T
T	F	T	F	F	F
T	T	F	F	T	T

The Biconditional Operator

- The biconditional $p \leftrightarrow q$ states that p *is true if and only if (IFF)* q is true.

Example

$p =$ "Bush wins the 2004 election."
$q =$ "Bush will be president for all of 2005."
$p \leftrightarrow q=$ "If, and only if, Bush wins the 2004 election, Bush will be president for all of 2005."



Biconditional Truth Table

- $p \leftrightarrow q$ means that p and q have the **same** truth value.
- Note this truth table is the exact **opposite** of $\oplus$'s!
  - $p \leftrightarrow q$ means $\neg(p \oplus q)$.
- $p \leftrightarrow q$ does **not** imply p and q are true, or cause each other.

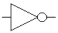
p	q	$p \leftrightarrow q$
F	F	T
F	T	F
T	F	F
T	T	T

Boolean Operations Summary

- We have seen 1 unary operator (out of the 4 possible) and 5 binary operators (out of the 16 possible). Their truth tables are below.

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \oplus q$	$p \rightarrow q$	$p \leftrightarrow q$
F	F	T	F	F	F	T	T
F	T	T	F	T	T	T	F
T	F	F	F	T	T	F	F
T	T	F	T	T	F	T	T

Some Alternative Notations

Name:	not	and	or	xor	implies	iff
Propositional logic:	$\neg$	$\wedge$	$\vee$	$\oplus$	$\rightarrow$	$\leftrightarrow$
Boolean algebra:	$\bar{p}$	pq	$+$	$\oplus$		
C/C++/Java (wordwise):	!	&&	||	!=		==
C/C++/Java (bitwise):	~	&	|	^		
Logic gates:						

Bits and Bit Operations

- A bit is a binary (base 2) digit: 0 or 1.
- Bits may be used to represent truth values.
- By convention: 0 represents "false"; 1 represents "true".
- Boolean algebra is like ordinary algebra except that variables stand for bits, + means "or", and multiplication means "and".
  - See chapter 10 for more details.
- Example (in C language):

End of §1.1

You have learned about:

- Propositions: What they are.
- Propositional logic operators'
  - Symbolic notations.
  - English equivalents.
  - Logical meaning.
  - Truth tables.
- Atomic vs. compound propositions.
- Alternative notations.
- Bits and bit-strings.
- Next section: §1.2
  - Propositional equivalences.
  - How to prove them.

§1.2 Propositional Equivalence

Propositional Equivalence

- Two *syntactically* (*i.e.*, textually) different compound propositions may be the *semantically* identical (i.e., have the same meaning). We call them equivalent.
- Learn:
  - Various *equivalence rules* or *laws*.
  - How to *prove* equivalences using *symbolic derivations*.

Tautologies and Contradictions

- A *tautology* is a compound proposition that is **true** *no matter what* the truth values of its atomic propositions are!

Example

$p \vee \neg p$ [What is its truth table?]

- A *contradiction* is a compound proposition that is **false** no matter what!

Example

$p \wedge \neg p$ [Truth table?]

- Other compound props. are *contingencies*.

Logical Equivalence

- Compound proposition p is *logically equivalent* to compound proposition q, written $p \iff q$, **IFF** the compound proposition $p \leftrightarrow q$ is a tautology.
- Compound propositions p and q are logically equivalent to each other **IFF** p and q contain the same truth values as each other in **all** rows of their truth tables.

Proving Equivalence via Truth Tables

Example

Prove that $p \vee q \Longleftrightarrow \neg(\neg p \wedge \neg q)$.

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

Equivalence Laws

- These are similar to the arithmetic identities you may have learned in algebra, but for propositional equivalences instead.
- They provide a pattern or template that can be used to match all or part of a much more complicated proposition and to find an equivalence for it.

Equivalence Laws - Examples

- Identity
  - $p \wedge \mathbf{T} \iff p$; $p \vee \mathbf{F} \iff p$.
- Domination
  - $p \vee \mathbf{T} \iff \mathbf{T}$; $p \wedge \mathbf{F} \iff \mathbf{F}$.
- Idempotent
  - $p \vee p \iff p$; $p \wedge p \iff p$.
- Double negation
  - $\neg\neg p \iff p$.
- Commutative
  - $p \vee q \iff q \vee p$; $p \wedge q \iff q \wedge p$.
- Associative
  - $(p \vee q) \vee r \iff p \vee (q \vee r)$.
  - $(p \wedge q) \wedge r \iff p \wedge (q \wedge r)$.

More Equivalence Laws

- Distributive
  - $p \vee (q \wedge r) \iff (p \vee q) \wedge (p \vee r)$.
  - $p \wedge (q \vee r) \iff (p \wedge q) \vee (p \wedge r)$.
- De Morgan's
  - $\neg(p \wedge q) \iff \neg p \vee \neg q$.
  - $\neg(p \vee q) \iff \neg p \wedge \neg q$.
- Trivial tautology/contradiction
  - $p \vee \neg p \iff \mathbf{T}$.
  - $p \wedge \neg p \iff \mathbf{F}$.

Defining Operators via Equivalences

- Using equivalences, we can define operators in terms of other operators.
  - Exclusive OR
    - $p \oplus q \Longleftrightarrow (p \vee q) \wedge (\neg p \vee \neg q)$
    - $p \oplus q \Longleftrightarrow (p \wedge \neg q) \vee (\neg p \wedge q)$
  - Implies
    - $p \rightarrow q \Longleftrightarrow \neg p \vee q$
  - Biconditional:
    - $p \leftrightarrow q \Longleftrightarrow (p \rightarrow q) \wedge (q \rightarrow p)$
    - $p \leftrightarrow q \Longleftrightarrow \neg(p \oplus q)$

An Example Problem

- Check using a symbolic derivation whether

$$(p \wedge \neg q) \rightarrow (p \oplus r) \iff \neg p \vee q \vee \neg r.$$

  - $(p \wedge \neg q) \rightarrow (p \oplus r)$
  - [Expand definition of$\rightarrow$]
    $\iff \neg(p \wedge \neg q) \vee (p \oplus r)$
  - [Definition of $\oplus$]
    $\iff \neg(p \wedge \neg q) \vee ((p \vee r) \wedge \neg(p \wedge r))$
  - [DeMorgan's Law]
    $\iff (\neg p \vee q) \vee ((p \vee r) \wedge \neg(p \wedge r))$
  - [$\vee$ commutes]
    $\iff (q \vee \neg p) \vee ((p \vee r) \wedge \neg(p \wedge r))$

- Example Continued...
  - [$\vee$ associative]
    $\Longleftrightarrow q \vee (\neg p \vee ((p \vee r) \wedge \neg(p \wedge r)))$
  - [distrib $\vee$ over $\wedge$]
    $\Longleftrightarrow q \vee (((\neg p \vee (p \vee r)) \wedge (\neg p \vee \neg(p \wedge r)))$
  - [assoc.]
    $\Longleftrightarrow q \vee (((\neg p \vee p) \vee r) \wedge (\neg p \vee \neg(p \wedge r)))$
  - [trivial taut.]
    $\Longleftrightarrow q \vee ((T \vee r) \wedge (\neg p \vee \neg(p \wedge r)))$
  - [domination]
    $\Longleftrightarrow q \vee (T \wedge (\neg p \vee \neg(p \wedge r)))$
  - [identity]
    $\Longleftrightarrow q \vee (\neg p \vee \neg(p \wedge r))$

- End of Long Example
    - [DeMorgan's]
      $\iff q \vee (\neg p \vee (\neg p \vee \neg r)$
    - [Assoc.]
      $\iff q \vee ((\neg p \vee \neg p) \vee \neg r)$
    - [Idempotent]
      $\iff q \vee (\neg p \vee \neg r)$
    - [Assoc.]
      $\iff (q \vee \neg p) \vee \neg r$
    - [Commut.]
      $\iff \neg p \vee q \vee \neg r$

Review: Propositional Logic(§1.1-§1.2)

- Atomic propositions: $p, q, r, \ldots$
- Boolean operators: $\neg \land \lor \oplus \rightarrow \leftrightarrow$
- Compound propositions: $s :\equiv (p \land \neg q) \lor r$
- Equivalences: $p \land \neg q \iff \neg(p \rightarrow q)$
- Proving equivalences using:
  - Truth tables.
  - Symbolic derivations. $p \iff q \iff r \ldots$

§1.3 Predicate Logic

Predicate Logic

- *Predicate logic* is an extension of propositional logic that permits concisely reasoning about whole *classes* of entities.
- Propositional logic (recall) treats simple *propositions* (sentences) as atomic entities.
- In contrast, *predicate* logic distinguishes the *subject* of a sentence from its *predicate*.
    - Remember these English grammar terms?

Subjects and Predicates

- In the sentence "The dog is sleeping":
  - The phrase "the dog" denotes the *subject* - the *object* or *entity* that the sentence is about.
  - The phrase "is sleeping" denotes the predicate - a property that is true of the subject.
- In predicate logic, a *predicate* is modeled as a *function* $P(\cdot)$ from objects to propositions.
  - $P(x) =$ "x is sleeping" (where x is any object).

More About Predicates

- Convention: Lowercase variables $x, y, z \ldots$ denote objects/entities; uppercase variables $P, Q, R \ldots$ denote propositional functions (predicates).
- Keep in mind that the *result of applying* a predicate P to an object x is the *proposition* $P(x)$. But the predicate P **itself** (e.g. $P =$"is sleeping") is not a proposition (not a complete sentence).

Example

If $P(x) =$"x is a prime number", then $P(3)$ is the *proposition* "3 is a prime number.".

Propositional Functions

- Predicate logic *generalizes* the grammatical notion of a predicate to also include propositional functions of **any** number of arguments, each of which may take **any** grammatical role that a noun can take.

Example

Let $P(x, y, z) =$ "x gave y the grade z".
If $x =$ "Mike", $y =$ "Mary", $z =$ "A", then $P(x, y, z) =$ "Mike gave Mary the grade A."

Universes of Discourse (U.D.s)

- The collection of values that a variable x can take is called x's *universe of discourse*.
- The power of distinguishing objects from predicates is that it lets you state things about *many* objects at once.

Example

Let $P(x) =$ "$x + 1 > x$".
We can then say, "For any number x, $P(x)$is true" instead of $(0 + 1 > 0) \wedge (1 + 1 > 1) \wedge (2 + 1 > 2)$...

Quantifier Expressions

- *Quantifiers* provide a notation that allows us to *quantify* (count) how many objects in the univ. of disc. satisfy a given predicate.
- "∀" is the FOR ∀LL or *universal* quantifier $\forall x P(x)$ means *for all* x in the u.d., P holds.
- "∃" is the ∃XISTS or *existential* quantifier $\exists x P(x)$ means there exists an x in the u.d. (that is, 1 or more) such that $P(x)$ is true.

The Universal Quantifier

Example

Let the u.d. of x be parking spaces at NCTU. Let $P(x)$ be the *predicate* "x is full." Then the *universal quantification* of $P(x)$, $\forall x\, P(x)$, is the *proposition*:

- "All parking spaces at NCTU are full."
- "Every parking space at NCTU is full."
- "For each parking space at NCTU, that space is full."

Examples of the Universal Quantifier

Example

Let $Q(x)$ be the statement "$x < 2$". What is the truth value of the quantification ∀x Q(x), where the domain consists of all real numbers?

Example

Suppose that $P(x)$ is "$x^2 > 0$". Show that the statement $\forall x P(x)$ is false where the universe of discourse consists of all integers.

Problem

What is the outcome if the domain is an empty set?

The Existential Quantifier

Example

Let the u.d. of x be parking spaces at NCTU. Let $P(x)$ be the *predicate* "x is full." Then the *existential quantification* of $P(x)$, $\exists x P(x)$, is the *proposition*:

- "Some parking space at NCTU is full."
- "There is a parking space at NCTU that is full."
- "At least one parking space at NCTU is full."

Examples of the Existential Quantifier

Example

Let $P(x)$ denote the statement "$x > 2$". What is the truth value of the quantification $\exists x P(x)$, where the domain consists of all real numbers?

Example

Suppose that $Q(x)$ is "$x = x + 1$". What is the truth value of the quantification $\exists x Q(x)$, where the domain consists of all real numbers?

Problem

What is the outcome if the domain is an empty set?

Free and Bound Variables

- An expression like $P(x)$ is said to have a *free variable* x (meaning, x is undefined).
- A quantifier (either $\forall$ or $\exists$) *operates* on an expression having one or more free variables, and *binds* one or more of those variables, to produce an expression having one or more *bound* variables.

Example of Binding

- $P(x, y)$ has 2 free variables, x and y.
- $\forall x P(x, y)$ has 1 free variable, and one bound variable. [Which is which?]
- "$P(x)$, where $x = 3$" is another way to bind x.
- An expression with **zero** free variables is a bona-fide (actual) proposition.
- An expression with **one or more** free variables is still only a predicate: $\forall x P(x, y)$.

Nesting of Quantifiers

Example

Let the u.d. of x & y be people.
Let $L(x, y) =$ "x likes y" (a predicate w. 2 f.v.'s)

- Then $\exists y L(x, y) =$ "There is someone whom x likes." (A predicate w. 1 free variable, x)
- Then $\forall x(\exists y L(x, y)) =$ "Everyone has someone whom they like." (A **Proposition** with **0** free variables.)

Quantifier Exercise

If $R(x, y) =$ "x relies upon y," express the following in unambiguous English:

- $\forall x(\exists y\ R(x, y))$
  - Everyone has some one to rely on.
- $\exists y(\forall x\ R(x, y))$
- $\exists x(\forall y\ R(x, y))$
- $\forall y(\exists x\ R(x, y))$
- $\forall x(\forall y\ R(x, y))$

Quantifier Exercise

If $R(x, y) =$ "x relies upon y," express the following in unambiguous English:

- $\forall x(\exists y\ R(x, y))$
  - Everyone has some one to rely on.
- $\exists y(\forall x\ R(x, y))$
  - There's a poor overburdened soul whom everyone relies upon (including himself)!
- $\exists x(\forall y\ R(x, y))$
- $\forall y(\exists x\ R(x, y))$
- $\forall x(\forall y\ R(x, y))$

Quantifier Exercise

If $R(x, y) =$ "x relies upon y," express the following in unambiguous English:

- $\forall x(\exists y\ R(x, y))$
  - Everyone has some one to rely on.
- $\exists y(\forall x\ R(x, y))$
  - There's a poor overburdened soul whom everyone relies upon (including himself)!
- $\exists x(\forall y\ R(x, y))$
  - There's some needy person who relies upon everybody (including himself).
- $\forall y(\exists x\ R(x, y))$
- $\forall x(\forall y\ R(x, y))$

Quantifier Exercise

If $R(x, y) =$ "x relies upon y," express the following in unambiguous English:

- $\forall x(\exists y\ R(x, y))$
  - Everyone has some one to rely on.
- $\exists y(\forall x\ R(x, y))$
  - There's a poor overburdened soul whom everyone relies upon (including himself)!
- $\exists x(\forall y\ R(x, y))$
  - There's some needy person who relies upon everybody (including himself).
- $\forall y(\exists x\ R(x, y))$
  - Everyone has someone who relies upon them.
- $\forall x(\forall y\ R(x, y))$

Quantifier Exercise

If $R(x, y) =$ "x relies upon y," express the following in unambiguous English:

- $\forall x(\exists y\ R(x, y))$
  - Everyone has some one to rely on.
- $\exists y(\forall x\ R(x, y))$
  - There's a poor overburdened soul whom everyone relies upon (including himself)!
- $\exists x(\forall y\ R(x, y))$
  - There's some needy person who relies upon everybody (including himself).
- $\forall y(\exists x\ R(x, y))$
  - Everyone has someone who relies upon them.
- $\forall x(\forall y\ R(x, y))$
  - Everyone relies upon everybody, (including themselves)!

Natural language is ambiguous!

- "Everybody likes somebody."
  - For everybody, there is somebody they like,
    - $\forall x \exists y \ \text{Likes}(x, y)$ *[Probably more likely.]*
  - or, there is somebody (a popular person) whom everyone likes?
    - $\exists y \forall x \ \text{Likes}(x, y)$
- "Somebody likes everybody."
  - Same problem: Depends on context, emphasis.

Still More Conventions

- Sometimes the universe of discourse is restricted within the quantification,
  - $\forall x > 0, P(x)$ is shorthand for
    "For all x that are greater than zero, $P(x)$."
    $\forall x(x > 0 \rightarrow P(x))$

Example

- $\exists x > 0, P(x)$ is shorthand for
  "There is an x greater than zero such that $P(x)$."
  $\exists x(x > 0 \wedge P(x))$

More to Know About Binding

- $\forall x \exists x P(x)$ – x is not a free variable in $\exists x P(x)$, therefore the $\forall x$ *binding isn't used*.
- $(\forall x P(x)) \wedge Q(x)$ – The variable x is outside of the scope of the $\forall x$ quantifier, and is therefore free. Not a proposition!
- $(\forall x P(x)) \wedge (\exists x Q(x))$ – This is legal, because there are 2 *different* x's!

Quantifier Equivalence Laws

- Definitions of quantifiers: If u.d.$= a, b, c, \ldots$
  - $\forall x P(x) \Leftrightarrow P(a) \wedge P(b) \wedge P(c) \wedge \ldots$
  - $\exists x P(x) \Leftrightarrow P(a) \vee P(b) \vee P(c) \vee \ldots$
- Negations of Quantified Expressions
  - $\neg \forall x P(x) \Leftrightarrow \exists x \neg P(x)$
  - $\neg \exists x P(x) \Leftrightarrow \forall x \neg P(x)$
- From those, we can prove the laws:
  - $\forall x\ P(x) \Leftrightarrow \neg(\exists x\ \neg P(x))$
  - $\exists x\ P(x) \Leftrightarrow \neg(\forall x\ \neg P(x))$
- Which propositional equivalence laws can be used to prove this?

DeMorgan's

More Equivalence Laws

- $\forall x \forall y\ P(x, y) \Leftrightarrow \forall y \forall x\ P(x, y)$
  $\exists x \exists y\ P(x, y) \Leftrightarrow \exists y \exists x\ P(x, y)$
- $\forall x (P(x) \wedge Q(x)) \Leftrightarrow (\forall x P(x)) \wedge (\forall x Q(x))$
  $\exists x (P(x) \vee Q(x)) \Leftrightarrow (\exists x P(x)) \vee (\exists x Q(x))$

Example

See if you can prove these yourself.
What propositional equivalences did you use?

Examples of Negations

- What are the negations of the statement “There is an honest politician” and “All Americans eat cheeseburgers”?
- What are the negations of the statements $\forall x(x^2 > x)$ and $\exists x(x^2 = 2)$?

Review: Predicate Logic (§1.3)

- Objects $x, y, z, \ldots$
- Predicates $P, Q, R, \ldots$ are functions mapping objects x to propositions $P(x)$.
- Multi-argument predicates $P(x, y)$.
- Quantifiers:
  - $[\forall x (P(x)] :\equiv$ "For all x's, $P(x)$."
  - $[\exists x (P(x)] :\equiv$ "There is an x such that $P(x)$."
- Universes of discourse, bound & free vars.

More Notational Conventions

- Quantifiers bind as loosely as needed: parenthesize $\forall x(P(x) \wedge Q(x))$
- Consecutive quantifiers of the same type can be combined: $\forall x \forall y \forall z P(x, y, z) \Leftrightarrow \forall x, y, z\ P(x, y, z)$ or even $\forall xyz\ P(x, y, z)$.

Defining New Quantifiers

- As per their name, quantifiers can be used to express that a predicate is true of any given *quantity* (number) of objects.
- Define $\exists! x P(x)$ to mean "$P(x)$ is true of *exactly one* x in the universe of discourse."
- $\exists! x P(x) \Leftrightarrow \exists x (P(x) \land \nexists y (P(y) \land y \neq x))$ "There is an x such that $P(x)$, where there is no y such that $P(y)$ and y is other than x."

Some Number Theory Examples

Example

Let u.d. $=$ the natural numbers $0, 1, 2, \ldots$

- "A number x is even, $E(x)$, if and only if it is equal to 2 times some other number."
  $\forall x(E(x) \leftrightarrow (\exists y \text{ s.t. } x = 2y))$
- "A number is prime, $P(x)$, iff it's greater than 1 and it isn't the product of two non-unity numbers."
  $\forall x(P(x) \leftrightarrow (x > 1 \land \neg\exists yz\,(x = yz \land y \neq 1 \land z \neq 1)))$

Goldbach's Conjecture (Unproven)

- Using $E(x)$ and $P(x)$ from previous slide,
  $\forall E(x > 2) : \exists P(p), P(q) : p + q = x.$
- or, with more explicit notation:
  $\forall x [x > 2 \wedge E(x)] \rightarrow \exists p \exists q P(p) \wedge P(q) \wedge p + q = x.$
- "Every even number greater than 2 is the sum of two primes."

Calculus Example

- One way of precisely defining the calculus concept of a limit, using quantifiers:

$$\left(\lim_{x \to a} f(x) = L\right)$$
$$\Leftrightarrow \left(\begin{array}{c} \forall \varepsilon > 0 : \exists \delta > 0 : \forall x : \\ (|x - a| < \delta) \to (|f(x) - L|) < \varepsilon \end{array}\right)$$

End of §1.3-§1.4, Predicate Logic

- From these sections you should have learned:
  - Predicate logic notation & conventions
  - Conversions: predicate logic $\leftrightarrow$ clear English
  - Meaning of quantifiers, equivalences
  - Simple reasoning with quantifiers
- Upcoming topics:
  - Introduction to proof-writing.
  - Then: Set theory –
    - a language for talking about collections of objects.