Discrete Mathematics
Lec 1: Logic and Proofs:Part 1

馬誠佑

What is Discrete Mathematics?

Discrete mathematics is the part of mathematics devoted to the study of discrete (as opposed to continuous) objects.

Calculus deals with continuous objects and is not part of discrete mathematics.

Examples of discrete objects: integers, steps taken by a computer program, distinct paths to travel from point A to point B on a map along a road network, ways to pick a winning set of numbers in a lottery.

A course in discrete mathematics provides the mathematical background needed for all subsequent courses in computer science and for all subsequent courses in the many branches of discrete mathematics.

Kinds of Problems Solved Using Discrete Mathematics [1]

How many ways can a password be chosen following specific rules?

How many valid Internet addresses are there?

What is the probability of winning a particular lottery?

Is there a link between two computers in a network?

How can I identify spam email messages?

How can I encrypt a message so that no unintended recipient can read it?

How can we build a circuit that adds two integers?

Kinds of Problems Solved Using Discrete Mathematics 2

What is the shortest path between two cities using a transportation system?

Find the shortest tour that visits each of a group of cities only once and then ends in the starting city.

How can we represent English sentences so that a computer can reason with them?

How can we prove that there are infinitely many prime numbers?

How can a list of integers be sorted so that the integers are in increasing order?

How many steps are required to do such a sorting?

How can it be proved that a sorting algorithm always correctly sorts a list?

Goals of a Course in Discrete Mathematics [1]

Mathematical Reasoning: Ability to read, understand, and construct mathematical arguments and proofs.

Combinatorial Analysis: Techniques for counting objects of different kinds.

Discrete Structures: Abstract mathematical structures that represent objects and the relationships between them. Examples are sets, permutations, relations, graphs, trees, and finite state machines.

Goals of a Course in Discrete Mathematics 2

Algorithmic Thinking: One way to solve many problems is to specify an algorithm. An algorithm is a sequence of steps that can be followed to solve any instance of a particular problem. Algorithmic thinking involves specifying algorithms, analyzing the memory and time required by an execution of the algorithm, and verifying that the algorithm will produce the correct answer.

Applications and Modeling: It is important to appreciate and understand the wide range of applications of the topics in discrete mathematics and develop the ability to develop new models in various domains. Concepts from discrete mathematics have not only been used to address problems in computing, but have been applied to solve problems in many areas such as chemistry, biology, linguistics, geography, business, etc.

Discrete Mathematics is a Gateway Course

Topics in discrete mathematics will be important in many courses that you will take in the future:

- **Computer Science**: Computer Architecture, Data Structures, Algorithms, Programming Languages, Compilers, Computer Security, Databases, Artificial Intelligence, Networking, Graphics, Game Design, Theory of Computation, ……
- **Mathematics**: Logic, Set Theory, Probability, Number Theory, Abstract Algebra, Combinatorics, Graph Theory, Game Theory, Network Optimization, …
  - The concepts learned will also be helpful in continuous areas of mathematics.
- **Other Disciplines**: You may find concepts learned here useful in courses in philosophy, economics, linguistics, and other departments.

Proposition (命題)

- Def:

A declaration sentence that is either true or false, but not both.

The truth value: True (T) or False (F)

Example

- Which of the following statements are propositions ? What are their truth value ?

1. Washington, D.C. is the capital of US.
2. Toronto is the capital of Canada.
3. 1+1 = 2
4. 2+2 = 3
5. What time is it ?
6. Read it carefully.
7. X+12 = 13
8. X+Y = Z

Propositional Variable

- Variables that represent propositions.
  e.g. p, q, r, s… (習慣上用小寫)

- The area of logic that deals with propositions:
  propositional calculus (命題演算、命題計算)

Compound Propositions (複合命題)

- Def:

Propostions that are formed from existing propostions using "Logical Operators".

Logical Operators: Negation

Negation Operator **($\neg$ *or* $\overline{}$)**

e.g. Let p be a proposition. $\boldsymbol{\neg p}$ ***or*** $\boldsymbol{\overline{p}}$ **(is read "not $\boldsymbol{p}$"),** The negation of p, is the statement "It's not the case of p"

The truth value of $\neg p$ is the opposite of the truth value of p.

e.g. p = "Taipei is the capital of R.O.C."

$\neg p$ = "It is not the case of Taipei is the capital of R.O.C."

= "Taipei is not the capital of R.O.C."

Truth table (真值表)

p	$\boldsymbol{\neg p}$
T	F
F	T

Logical Operators: Conjunction Operator ($\wedge$, AND)

Conjunction Operator ($\wedge$, AND)

e.g. Let p and q be propositions. $p \wedge q$ is true when p and q are both true or false otherwise.

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

Logical Operators: Disjunction Operator ($\vee$, OR)

Disjunction Operator ($\vee$, OR)

e.g. Let p and q be propositions. $p \vee q$, the disjunction of p and q, is the proposition "p or q" .

$p \vee q$ is false when p and q are false and true otherwise.

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

Example

p = "There are over 16GB unused space in the HD of Rebecca's computer."

q = "The processor in Rebecca's computer operates at a speed greater than 1GHz."

r = "There is a turtle on Rebecca's table."

$p \wedge q$ = "There are over 16GB unused space in the HD of Rebecca's computer and the processor in Rebecca's computer operates at a speed greater than 1GHz."

$p \vee q$= "…"

$p \wedge q \wedge r$ = "…"

$p \vee q \vee r$= "…"

Logical Operators: Exclusive OR ($\oplus$ 互斥)

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

Conditional Operator (→, imply)

$p \to q$ conditional statement. (若p則q)

When p is true and q is false, $p \to q$ is false

e.g. “If you got 100 score in your final exam, your grade of the semester m must be A” (p:前提、前件、假設)

e.g. If p denotes “I am at home.” and q denotes “It is raining.” then $p \to q$ denotes “If I am at home then it is raining.”

In $p \to q$, p is the *hypothesis* (*antecedent* or *premise*) and q is the *conclusion* (or *consequence*).

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

Understanding Implication [1]

In $p \rightarrow q$ there does not need to be any connection between the antecedent or the consequent. The "meaning" of $p \rightarrow q$ depends only on the truth values of p and q.

These implications are perfectly fine, but would not be used in ordinary English.

- "If the moon is made of green cheese, then I have more money than Bill Gates. "
- "If the moon is made of green cheese then I'm on welfare."
- "If 1 + 1 = 3, then your grandma wears combat boots."

Understanding Implication 2

One way to view the logical conditional is to think of an obligation or contract.

- “If I am elected, then I will lower taxes.”
- “If you get 100% on the final, then you will get an A.”

If the politician is elected and does not lower taxes, then the voters can say that he or she has broken the campaign pledge. Something similar holds for the professor. This corresponds to the case where p is true and q is false.

Different Ways of Expressing $p \to q$

if p, then q

if p, q

q unless $\neg p$

q if p

q whenever p

q follows from p

p implies q

p only if q

q when p

p is sufficient for q

q is necessary for p

a necessary condition for p is q

a sufficient condition for q is p

Compound Expressions

Connectives from propositional logic carry over to predicate logic.

If *P(x)* denotes "$x > 0$," find these truth values:

$P(3) \vee P(-1)$ **Solution**: T

$P(3) \wedge P(-1)$ **Solution**: F

$P(3) \rightarrow P(-1)$ **Solution**: F

$P(3) \rightarrow \neg P(-1)$ **Solution**: T

Expressions with variables are not propositions and therefore do not have truth values. For example,

$P(3) \wedge P(y)$

$P(x) \rightarrow P(y)$

When used with quantifiers (to be introduced next), these expressions (propositional functions) become propositions.

Exercise

- If 3+4 = 6, then x = x +1,

  If 3+2 = 5, then x = x +1

$p\text{:}\ 3+4 = 6$

$q\text{:}\ 3+2 = 5$

$r\text{:}\ x=x+1$

$p \rightarrow r$?

$q \rightarrow r$?

Biconditional Operator ($\leftrightarrow$)

$$p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$$

If p and q are propositions, then we can form the *biconditional* proposition $p \leftrightarrow q$, read as "p if and only if q." The biconditional $p \leftrightarrow q$ denotes the proposition with this truth table:

p	q	$p \to q$	$q \to p$	$(p \to q) \wedge (q \to p)$
T	T	T	T	T
T	F	F	T	F
F	T	T	F	F
F	F	T	T	T

e.g. If p denotes "I am at home." and q denotes "It is raining." then $p \leftrightarrow q$ denotes "I am at home if and only if it is raining."

Expressing the Biconditional

Some alternative ways “*p* if and only if *q*” is expressed in English:

- *p* is necessary and sufficient for *q*
- if *p* then *q*, and conversely
- *p* if *q*

•當命題「若P則Q」為真時，P稱為Q的**充分條件**，Q稱為P的**必要條件**。
因此：
•當命題「若P則Q」與「若Q則P」皆為真時，P是Q的**充分必要條件**，同時，Q也是P的**充分必要條件**。
•當命題「若P則Q」為真，而「若Q則P」為假時，我們稱P是Q的**充分不必要條件**，Q是P的**必要不充分條件**，反之亦然。

Converse, Contrapositive, and Inverse

- If p ,then q ($p \rightarrow q$) 如果p，則q; p蘊含q; 只有在q時，才可能p; p是q的充分條件; q是p的必要條件; 每當p發生， q就發生; 沒有p，就沒有q; q從p推得; 除非$\neg \boldsymbol{p}$，否則q 。(implication)
- p if and only if q $(p \leftrightarrow q)$ 若且為若; p是q的充要條件; 若p則q，並且反之亦然; p iff q 。
- $q \rightarrow p$為$p \rightarrow q$的**逆命題; 換位命題(converse)**
- $\neg p \rightarrow \neg q$ 為$p \rightarrow q$的**反命題; 異質命題 (inverse)**
- $\neg q \rightarrow \neg p$為$p \rightarrow q$的**質位互換命題 (contrapositive)**

Converse, Contrapositive, and Inverse: Example

- **Example**: Find the converse, inverse, and contrapositive of “It raining is a sufficient condition for my not going to town.”

- **Solution:**

  converse: If I do not go to town, then it is raining. ($q \rightarrow p$為$p \rightarrow q$的逆命題)

  inverse: If it is not raining, then I will go to town. ($\neg p \rightarrow \neg q$ 為$p \rightarrow q$的反命題)

  contrapositive: If I go to town, then it is not raining. ($\neg q \rightarrow \neg p$為$p \rightarrow q$的質位互換命題)

Operator Priority

Operator	Priority
¬	1
∧	2
∨	3
→	4
↔	5

p	q	r	$p \vee q$	$p \wedge r$	$p \vee q \rightarrow p \wedge r$
T	T	T	T	T	T
T	T	F	T	F	F
T	F	T	T	T	T
T	F	F	T	F	F
F	T	T	T	F	F
F	T	F	T	F	F
F	F	T	F	F	T
F	F	F	F	F	T

Propositional Equivalences

- Def:

  A compound proposition is always **true** (or **false**, resp.) is called a **tautology** 恆真式(or **contradiction** 矛盾, resp.)

- A compound proposition is neither a **tautology** nor a **contradiction** is called a **contingency**偶然性.

Truth Tables For Compound Propositions

Construction of a truth table:

Rows

- Need a row for every possible combination of values for the atomic propositions.

Columns

- Need a column for the compound proposition (usually at far right)
- Need a column for the truth value of each expression that occurs in the compound proposition as it is built up.
  - This includes the atomic propositions

Example

- Let p be a proposition.

$\boldsymbol{p \wedge \neg p, p \vee \neg p}$

$\boldsymbol{p}$	$\boldsymbol{\neg p}$	$\boldsymbol{p \wedge \neg p}$	$\boldsymbol{p \vee \neg p}$
T	F	F	T
F	T	F	T

Logical Equivalence

- Def:

Two compound propositions $\boldsymbol{p}$ and $\boldsymbol{q}$ are called logical equivalent if $p \longleftrightarrow q$ tautology, and is denoted as $p \equiv q$

Example

- $p \rightarrow q, \neg q \rightarrow \neg p$

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

$(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$

De Morgan Laws

$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

$$\neg(p \lor q) \equiv \neg p \land \neg q \qquad \neg(p \land q) \equiv \neg p \lor \neg q$$

p	q	$\neg p$	$\neg q$	$p \lor q$	$\neg(p \lor q)$	$\neg p \land \neg q$	$\neg(p \lor q) \leftrightarrow \neg p \land \neg q$
T	T	F	F	T	F	F	T
T	F	F	T	T	F	F	T
F	T	T	F	T	F	F	T
F	F	T	T	F	T	T	T

結合律 （Associative property）

- $(\boldsymbol{p} \wedge \boldsymbol{q}) \wedge \boldsymbol{r} \equiv \boldsymbol{p} \wedge (\boldsymbol{q} \wedge \boldsymbol{r})$

p	q	r	$p \wedge q$	$q \wedge r$	$(p \wedge q) \wedge r$	$p \wedge (q \wedge r)$
T	T	T				
T	T	F				
T	F	T				
T	F	F				
F	T	T				
F	T	F				
F	F	T				
F	F	F				

交換律 （Commutative property）

- $p \wedge q \equiv q \wedge p$
- $\boldsymbol{(p \vee q) \vee r \equiv p \vee (q \vee r) \equiv p \vee q \vee r}$
- $\boldsymbol{p \vee q \equiv q \vee p}$

- Identitiy laws: $p \wedge \mathrm{T} \equiv p$

  $p \vee \mathrm{F} \equiv p$
- Domination laws: $p \vee \mathrm{T} \equiv \mathrm{T}$

  $p \wedge \mathrm{F} \equiv \mathrm{F}$
- Idempotent laws: $p \wedge p \equiv p$

  $p \vee p \equiv p$
- Double negation: $\neg(\neg \boldsymbol{p}) \equiv \boldsymbol{p}$
- Commutative laws: $\boldsymbol{p} \vee \boldsymbol{q} \equiv \boldsymbol{q} \vee \boldsymbol{p}$

  $p \wedge q \equiv q \wedge p$
- Associative laws: $(\boldsymbol{p} \vee \boldsymbol{q}) \vee \boldsymbol{r} \equiv \boldsymbol{p} \vee (\boldsymbol{q} \vee \boldsymbol{r}) \equiv \boldsymbol{p} \vee \boldsymbol{q} \vee \boldsymbol{r}$

  $(\boldsymbol{p} \wedge \boldsymbol{q}) \wedge \boldsymbol{r} \equiv \boldsymbol{p} \wedge (\boldsymbol{q} \wedge \boldsymbol{r}) \equiv \boldsymbol{p} \wedge \boldsymbol{q} \wedge \boldsymbol{r}$
- Distributive laws: $\boldsymbol{p} \wedge (\boldsymbol{q} \vee \boldsymbol{r}) \equiv (\boldsymbol{p} \wedge \boldsymbol{q}) \vee (\boldsymbol{p} \wedge r)$

  $\boldsymbol{p} \vee (\boldsymbol{q} \wedge \boldsymbol{r}) \equiv (\boldsymbol{p} \vee \boldsymbol{q}) \wedge (\boldsymbol{p} \vee r)$
- De Morgan laws: $\neg (\boldsymbol{p} \wedge \boldsymbol{q}) \equiv \neg \boldsymbol{p} \vee \neg \boldsymbol{q}$

  $\neg (\boldsymbol{p} \vee \boldsymbol{q}) \equiv \neg \boldsymbol{p} \wedge \neg \boldsymbol{q}$

- Absorption laws: $p \vee (p \wedge q) \equiv p$

  $p \wedge (p \vee q) \equiv p$
- Negation: $p \vee \neg p \equiv \mathrm{T}$

  $p \wedge \neg p \equiv \mathrm{F}$

$$P_1 \land P_2 \ldots \land P_n \equiv \bigwedge_{i=1}^{n} P_i$$

$$P_1 \lor P_2 \ldots \lor P_n \equiv \bigvee_{i=1}^{n} P_i$$

$$\neg\left(\bigwedge_{i=1}^{n} P_i\right) \equiv \bigvee_{i=1}^{n} \neg P_i$$

$$\neg\left(\bigvee_{i=1}^{n} P_i\right) \equiv \bigwedge_{i=1}^{n} \neg P_i$$

Conditional Statement

- $p \rightarrow q \equiv \neg q \rightarrow \neg p$
- $p \rightarrow q \equiv \neg p \vee q$
- $(p \rightarrow q) \wedge (p \rightarrow r) \equiv p \rightarrow (q \wedge r)$
- $(p \rightarrow r) \wedge (q \rightarrow r) \equiv (p \vee q) \rightarrow r$

Bi-conditional Statement

- $p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$
- $p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
- $p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$

代數運算

- Ex: $\neg(\boldsymbol{p} \rightarrow \boldsymbol{q}) \equiv \boldsymbol{p} \wedge \neg \boldsymbol{q}$
- Pf: (1) Truth table

  (2) $\neg(\boldsymbol{p} \rightarrow \boldsymbol{q}) \equiv \neg(\neg \boldsymbol{p} \vee \boldsymbol{q})$

  $\equiv \neg(\neg \boldsymbol{p}) \wedge \neg q$

  $\equiv \boldsymbol{p} \wedge \neg q$

Example

Ex:$\neg(\boldsymbol{p} \vee (\neg \boldsymbol{p} \wedge \boldsymbol{q})) \equiv \neg\, \boldsymbol{p} \wedge \boldsymbol{q}$

Pf:

$\neg(\boldsymbol{p} \vee (\neg \boldsymbol{p} \wedge \boldsymbol{q})) \equiv \neg[(p \vee (\neg \boldsymbol{p}) \wedge (p \vee q)]$

$\equiv \neg[T \wedge (p \vee q)] \equiv \neg\, (p \vee q) \equiv \neg\, p \wedge \neg\, q$

Example

- Ex: Prove that $(\boldsymbol{p} \wedge \boldsymbol{q}) \rightarrow (\boldsymbol{p} \vee \boldsymbol{q})$ **is a tautology**

Pf:

$(\boldsymbol{p} \wedge \boldsymbol{q}) \rightarrow (\boldsymbol{p} \vee \boldsymbol{q}) \equiv \neg (\boldsymbol{p} \wedge \boldsymbol{q}) \vee (\boldsymbol{p} \vee \boldsymbol{q})$

$\equiv (\neg \boldsymbol{p} \vee \neg \boldsymbol{q}) \vee (\boldsymbol{p} \vee \boldsymbol{q}) \equiv \neg \boldsymbol{p} \vee \neg \boldsymbol{q} \vee \boldsymbol{p} \vee \boldsymbol{q}$

$\equiv (\neg \boldsymbol{p} \vee \boldsymbol{p}) \vee (\neg \boldsymbol{q} \vee \boldsymbol{q}) \equiv \mathbf{T}$

Introducing Predicate Logic

Predicate logic uses the following new features:

- Variables: x, y, z
- Predicates: $P(x)$, $M(x)$
- Quantifiers (*to be covered in a few slides*):

Propositional functions are a generalization of propositions.

- They contain variables and a predicate, e.g., $P(x)$
- Variables can be replaced by elements from their *domain*.

Predicate Logic

- The area of logic that deals with predicates and quantifiers is called the predicate calculus.
- John is a boy.
  - "John" is subject.
  - "is a boy" is predicate.
- $P(x) = \text{“}x\ is\ a\ boy.\text{”}$
  - $P\ is\ a\ proposition\ function$.
  - $P(x)$ is the value of the proposition P at x.
  - For example, $P(Amy) = \text{“}Amy\ is\ a\ boy.\text{”}$
  - Note that $P(x)$ is not a proposition.

Propositional Functions

Propositional functions become propositions (and have truth values) when their variables are each replaced by a value from the *domain* (or *bound* by a quantifier, as we will see later).

The statement *P(x)* is said to be the value of the propositional function *P* at *x*.

For example, let *P(x)* denote "$x > 0$" and the domain be the integers. Then:

P(-3) is false.

P(0) is false.

P(3) is true.

Often the domain is denoted by *U*. So in this example *U* is the integers.

Examples of Propositional Functions

Let "$x + y = z$" be denoted by $R(x, y, z)$ and U (for all three variables) be the integers. Find these truth values:

R(2,-1,5)

Solution: F

R(3,4,7)

Solution: T

R(x, 3, z)

Solution: Not a Proposition

Now let "$x - y = z$" be denoted by $Q(x, y, z)$, with U as the integers. Find these truth values:

Q(2,-1,3)

Solution: T

Q(3,4,7)

Solution: F

Q(x, 3, z)

Solution: Not a Proposition

Examples of Propositional Functions

Ex. Let $\mathrm{P}(x) = "x > 3". Then,$

- P(2) = “2>3”, and the truth value of P(2) is false.
- P(4) = “4>3”, and the truth value of P(4) is true.

- A statement of the form P($x_1, x_2, \ldots, x_n$) is the value of the proposition function P at the n-tuple ($x_1, x_2, \ldots, x_n$), and P is also called an n-tuple predicate or an n-ary prediction

Universal Quantifier (∀)

- The universal quantification of P(x) is the statement "P(x) for all value of x in the domain."

- $\forall\ is\ called\ the\ universal\ quantifier$.
- $\forall x\ P(x)$ is read "for all x P(x)" or "for every x P(x)."
- An element for which P(x) is false is called a counter example of $\forall x\ P(x)$.
- Key concept: the domain, the domain of discourse, and the universal of discourse.

Example of the Universal Quantifier

- P(x) = “x is begin with B.” and U={Pineapple, Apple, Banana, Cherry}
  - $\forall x\ P(x) \equiv P(Pineapple) \wedge P(Apple) \wedge P(\ Banana) \wedge P(\ Cherry) \equiv \bigwedge_{x \in U} P(x)$.

Ex. Let P(x) = “x+1>x”. What is the truth value of the quantification $\forall x$ P(x) where the domain consists of all real number?

Existential Quantifier

- Def:

  The existential quantification of P(x) is the statement "There exists an element x in the domain such that P(x)."

- ∃ is called the existential quantifier.
- ∃x P(x) is read "There exists an element x in the domain such that P(x)."
- The statement is false if and only if $\forall x \neg P(x)$, i.e.,

$\neg(\exists x\ P(x)) \equiv \forall x\ \neg P(x)$.

- Ex. P(x) = "x is begin with B." and U={Pineapple, Apple, Banana, Cherry}
  - $\exists\ x\ P(x) \equiv P(Pineapple) \vee P(Apple) \vee P(\ Banana) \vee P(\ Cherry) \equiv \bigvee_{x \in U} P(x)$.

Existential Quantifier

$\exists x\, P(x)$ is read as "For some x, P(x)", or as "There is an x such that P(x)," or "For at least one x, P(x)."

Examples:

1. If $P(x)$ denotes "$x > 0$" and U is the integers, then $\exists x\, P(x)$ is true. It is also true if U is the positive integers.
2. If $P(x)$ denotes "$x < 0$" and U is the positive integers, then $\exists x\, P(x)$ is false.
3. If $P(x)$ denotes "x is even" and U is the integers, then $\exists x\, P(x)$ is true.

- $\exists x\ P(x) \equiv P(x_1) \vee P(x_2) \vee \ldots \vee P(x_n)$

$\neg\big(\exists x P(x)\big)$
$\equiv \neg(P(x_1) \vee P(x_2) \vee \ldots \vee P(x_n)\ )$
$\equiv \neg\ P(x_1) \wedge \neg P(x_2) \wedge \ldots \wedge \neg P(x_n)$
$\equiv \forall x(\neg P(x))$

- $\forall x\ P(x) \equiv P(x_1) \wedge P(x_2) \wedge \ldots \wedge\ P(x_n)$

$\neg\ \forall x\ P(x) \equiv \exists x(\neg P(x))$

Universal Quantifier v.s. Existential Quantifier

- True or False

Statement	When True	When False
$\forall x P(x)$	$P(x)$ is true for every x.	There is an x for which P(x) is false.
$\exists x P(x)$	There is an x for which P(x) is true.	P(x) is false for every x.

- The negation of quantifiers can be obtained by De Morgan laws
  - $\neg \forall x P(x) \equiv \exists x \neg P(x)$
  - $\neg \exists x P(x) \equiv \forall x \neg P(x)$

Examples

The truth value of $\exists x\, P(x)$ and $\forall x\, P(x)$ depend on both the propositional function $P(x)$ and on the domain U.

Examples:

1. If U is the positive integers and $P(x)$ is the statement "$x < 2$", then $\exists x\, P(x)$ is true, but $\forall x\, P(x)$ is false.

2. If U is the negative integers and $P(x)$ is the statement "$x < 2$", then both $\exists x\, P(x)$ and $\forall x\, P(x)$ are true.

3. If U consists of 3, 4, and 5, and $P(x)$ is the statement "$x > 2$", then both $\exists x\, P(x)$ and $\forall x\, P(x)$ are true. But if $P(x)$ is the statement "$x < 2$", then both $\exists x\, P(x)$ and $\forall x\, P(x)$ are false.

Precedence of Quantifiers

The quantifiers $\forall$ and $\exists$ have higher precedence than all the logical operators.

For example, $\forall x\, P(x) \vee Q(x)$ means $(\forall x\, P(x)) \vee Q(x)$

$\forall x\, (P(x) \vee Q(x))$ means something different.

Unfortunately, often people write $\forall x\, P(x) \vee Q(x)$ when they mean $\forall x\, (P(x) \vee Q(x))$.

Translating from English to Logic [1]

Example 1: Translate the following sentence into predicate logic: “Every student in this class has taken a course in Java.”

Solution:

First decide on the domain *U*.

- **Solution 1**: If *U* is all students in this class, define a propositional function J(x) denoting “x has taken a course in Java” and translate as $\forall x\ J(x)$.
- **Solution 2**: But if *U* is all people, also define a propositional function S(x) denoting “x is a student in this class” and translate as $\forall x\ (S(x) \rightarrow J(x))$.

$\forall x\ (S(x) \land J(x))$ is not correct. What does it mean?

Translating from English to Logic 2

Example 2: Translate the following sentence into predicate logic: “Some student in this class has taken a course in Java.”

Solution:

First decide on the domain *U*.

Solution 1: If *U* is all students in this class, translate as

$$\exists x\, J(x)$$

Solution 2: But if *U* is all people, then translate as

$$\exists x\, (S(x) \land J(x))$$

$\exists x\, (S(x) \rightarrow J(x))$ is not correct. What does it mean?

Translation from English to Logic

Examples:

1. "Some student in this class has visited Mexico."

   Solution: Let $M(x)$ denote "x has visited Mexico" and $S(x)$ denote "x is a student in this class," and U **be all people.**

$$\exists X\left(S(X) \wedge M(X)\right)$$

1. "Every student in this class has visited Canada or Mexico."

   Solution: Add $C(x)$ denoting "x has visited Canada."

$$\forall X\left(S(X) \rightarrow \left(M(X) V C(X)\right)\right)$$

Some Fun with Translating from English into Logical Expressions[3]

U = {fleegles, snurds, thingamabobs}

F(x): *x* is a fleegle

S(x): *x* is a snurd

T(x): *x* is a thingamabob

"All fleegles are snurds."

Solution: $\forall X\left(F(X) \rightarrow S(X)\right)$

Some Fun with Translating from English into Logical Expressions [4]

U = {fleegles, snurds, thingamabobs}

F(x): *x* is a fleegle

S(x): *x* is a snurd

T(x): *x* is a thingamabob

"Some fleegles are thingamabobs."

Solution: $\exists X\left(F(X)\wedge T(X)\right)$

Some Fun with Translating from English into Logical Expressions 5

U = {fleegles, snurds, thingamabobs}

F(x): *x* is a fleegle

S(x): *x* is a snurd

T(x): *x* is a thingamabob

"No snurd is a thingamabob."

Solution: $\neg\exists X\left(S(X)\wedge T(X)\right)$ **What is this equivalent to?**

Solution: $\forall X\left(\neg S(X)\vee\neg T(X)\right)$

Some Fun with Translating from English into Logical Expressions 6

U = {fleegles, snurds, thingamabobs}

F(x): x is a fleegle

S(x): *x* is a snurd

T(x): *x* is a thingamabob

"If any fleegle is a snurd then it is also a thingamabob."

Solution: $\forall X\left(\left(F(X) \wedge S(X)\right) \rightarrow T(X)\right)$