

Discrete Maths Notes

September 7, 2023

Contents

1	Logic	2
1.1	Propositional Logic	2
1.1.1	Basics	2
1.1.2	Terminology	2
1.1.3	Examples	3
1.1.4	Satisfiability, Tautology, Contradiction	3
1.2	Truth Tables	4
1.2.1	Basics	4
1.2.2	Connective Outputs	4
1.2.3	Examples	5
1.2.4	Exercise: Finding Tautologies, Satisfiable & Contradicting Props' .	5
1.3	Equivalence	7
1.3.1	Introduction to Equivalence	7
1.3.2	Equivalence Laws	9
1.4	Arguments	11
1.4.1	Valid Arguments / Inference Rules	12
1.4.2	Proving a Valid Argument	14
1.4.3	The Deduction Method	16
1.5	Predicate Logic	17
1.5.1	Predicate representation	18
1.5.2	Translating to Logical Expressions	20

1 Logic

1.1 Propositional Logic

1.1.1 Basics

A **proposition** is a statement that is either true or false.

Prepositions will be represented mathematically with capital letters A, B, C...

These prepositions are then are connected into more complex compound prepositions using *connectives*. Connectives are statements like "and, implies, if-then" and are represented mathematically with the symbols below.

❗ It's not always easy to determine if they're true/false.

Connectives			
Symbol	Name	English Term(s)	Reading
\wedge	AND	And, But, Also	A and B
\vee	OR	-	A or B
\implies	IMPLICATION	If A, Then B If A, then B A implies B A, therefore B A only if B B follows from A A is a sufficient condition for B B is a necessary condition for A	A implies B
\iff	BICONDITIONAL	If & only if A is necessary and sufficient for B	A if and only if B
\neg	NEGATION	Not...	Not A

❗ A Biconditional can also be thought of $(A \implies B) \wedge (B \implies A)$

Negation may sometimes be represented as A' or \overline{A}

1.1.2 Terminology

$A \wedge B$ - conjunction of conjuncts A and B

$A \vee B$ - disjunction of disjuncts A and B

$A \implies B$ - A is the hypothesis/antecedent and B is the conclusion/consequence

1.1.3 Examples

1 Compound Proposition

If all humans are mortal_{prp A} and all Greeks are human_{prp B}
then all Greeks are mortal_{prp C} can be represented as $A \wedge B \implies C$

2 Negation

Chocolate is sweet \rightarrow Chocolate is not sweet

Peter is tall and thin \rightarrow Peter is short or fat

The river is shallow or polluted \rightarrow The river is deep and polluted.

❗ Short and
fat would be
incorrect!

❗ Not shallow
or not pol-
luted would
be incorrect!

3 Implication: hypothesis and conclusion

If the rain continues then the river will flood

A sufficient condition for a network failure is that the central switch goes down

The avocados are ripe only if they are dark and soft

A good diet is a necessary condition for a healthy cat

1.1.4 Satisfiability, Tautology, Contradiction

A proposition is satisfiable if it is true for *at least one* combination of boolean values.

A Boolean Satisfiability Problem (SAT) is checking for satisfiability in a propositional logic formula.

❗ You don't
need a whole
truth table for
this, just look
for one!

A Tautology is a proposition that is always true

ex $A \vee \neg A$

A Contradiction is a proposition that is always false.

ex $A \wedge \neg A$

1.2 Truth Tables

1.2.1 Basics

Truth Tables are used for determining all the possible outputs of a complex compound proposition.

The Columns Are for the prepositions, intermediate compound prepositions and the whole compound proposition.

The Rows Are to contain the different sets of possible truth values for each proposition. You will have 2^p rows where p is the number of propositions (then +1 for the header).

ⓘ The intrmt³ prepositions are optional steps to make solving easier, use as needed.

▲ The connectives in a compound propositional logic problem follow an order of precedence (the PEMDAS of logic) in the following order;

\neg , \wedge , \vee , \implies , \iff

1.2.2 Connective Outputs

Negation	
A	$\neg A$
T	F
F	T

And		
A	B	$A \wedge B$
T	T	T
T	F	F
F	T	F
F	F	F

Or		
A	B	$A \vee B$
T	T	T
T	F	T
F	T	T
F	F	F

Implication		
A	B	$A \implies B$
T	T	T
T	F	F
F	T	T
F	F	T

Biconditional		
A	B	$A \iff B$
T	T	T
T	F	F
F	T	F
F	F	T

An implication is true when the hypothesis is false or the conclusion is true.

A Biconditional is true when the two propositions have the same value.

Out of all these outputs, the most unintuitive is the 3rd implication output ($F, T \implies T$). The easiest way to understand this output is with the proposition “If it is raining, then the ground is wet”; now say you step outside and it is not raining, but the ground is wet. The entire statement isn’t false or incorrect, but the first part of it still has a false value. The only way to make an implication false is when the hypothesis is true but the conclusion is false.

1.2.3 Examples

$$A \implies B \iff B \implies A$$

A	B	$A \implies B$	$B \implies A$	—
T	T	T	T	T
T	F	F	T	F
F	T	T	F	F
F	F	T	T	T

$$A \wedge \neg B \implies \neg C$$

A	B	C	$A \wedge \neg B$	—
T	T	T	F	T
T	T	F	F	T
T	F	T	T	F
T	F	F	T	T
F	T	T	F	T
F	T	F	F	T
F	F	T	F	T
F	F	F	F	T

i Remember, columns like $A \implies B$ are optional in-between steps to help solve each problem.

1.2.4 Exercise: Finding Tautologies, Satisfiable & Contradicting Props'

Indicate whether each of the following is a tautology, satisfiable but not a tautology or a contradiction;

$$A \implies B$$

$$A \implies A$$

$$A \implies \neg B \vee \neg C$$

$$A \vee B \implies B$$

$$(A \wedge B) \implies (A \vee B)$$

$$A \vee \neg A \implies B \wedge \neg B$$

(Answers and explanations on the next page...)

❶ Notice how none of these rely on drawing out a whole truth table! Focus on trying to find a way to get each proposition to output true and a way to get it to output false!

$$A \implies B$$

Satisfiable but not a tautology

Just knowing the properties of an implication you should know there's way to get true outputs and a false output.

$$A \implies A$$

Tautology

Only would be $T \implies T$ or $F \implies F$, both of which result in true.

$$A \implies \neg B \vee \neg C$$

Satisfiable but not a tautology

Instead of making a long unpleasant truth table, it's easiest to start by simply looking for one true and one false possible output.

We can make the left side true simply by making A false, since all that remains is an or statement we now have a true output.

We can just as easily find a false output for this proposition with $A = T$, $B = T$ ($\neg B = F$) to make the implication false, then we can just make $\neg C$ false to make the or output false.

$$A \vee B \implies B$$

Satisfiable but not a tautology

If we make B true then the biconditional will always be true regardless of A.

There is only one way to make an implication false, so if we can set up A and B to result in that false output, it won't be a tautology. If we make A true and B false it will make the implication false!

$$(A \wedge B) \implies (A \vee B)$$

Tautology

Remember the only way to make an implication false is if the hypothesis is true and the conclusion is false. There is absolutely no way to do this because of the and/or setup!


$$A \vee \neg A \implies B \wedge \neg B$$

Contradiction

The left side is always true and the right side is always false. So the result of the implication is always false!

1.3 Equivalence

1.3.1 Introduction to Equivalence

 Two (compound) propositions P and Q are **logically equivalent** when their truth values always match (Meaning they'll have the same truth table!). Equivalence is denoted by $P \equiv Q$.

Equivalence relates heavily to the concept of Tautologies;

P and Q are equivalent when $P \iff Q$ is a tautology.

A proposition P is a tautology iff (if and only if) it is equivalent to T (true), i.e $P \equiv T$

Examples

¹

Given the implication $A \implies B$, are the following equivalent?

The contrapositive: $\neg B \implies \neg A$

The converse: $B \implies A$


A	B	$A \implies B$	$\neg B \implies \neg A$	$B \implies A$
T	T	T	T	T
T	F	F	F	T
F	T	T	T	F
F	F	T	T	T

Looking at the table we can see that $A \implies B$ and $\neg B \implies \neg A$ are equivalent.

Now, what about $\neg A \vee B$?

A	B	$A \implies B$	$\neg A \vee B$
T	T	T	T
T	F	F	F
F	T	T	T
F	F	T	T

Yep! $\neg A \vee B \equiv A \implies B$.

 This is actually one of the equivalence laws you'll see in the next section!

Understanding equivalent boolean expressions is very important in computer science (for code) and chip design (for logic gates). Consider the code below;

```
if (x > 0 || (x <= 0 && y > 100))
```

Lets see if we can change this expression to something equivalent but simplified.

Let A be $x > 0$ and let B be $y > 100$

Now we can compare the truth values of $A \vee (\neg A \wedge B)$ and $A \vee B$.

A	B	$A \vee (\neg A \wedge B)$	$A \vee B$
T	T	T	T
T	F	T	T
F	T	T	T
F	F	F	F

They're equivalent! We can reduce the if statement's expression to simply;

```
if (x > 0 || y > 100)
```


1.3.2 Equivalence Laws

For more complex propositions it is impractical to create a set of massive truth tables to check for equivalence. So instead we utilize equivalence laws to directly prove equivalence.

Nine Equivalence Laws;

Many of these are pretty self-explanatory

Double Negation Law: $\neg(\neg A) \equiv A$

Identity Laws: $A \wedge T \equiv A$ $A \vee F \equiv A$

Domination Laws: $A \vee T \equiv T$ $A \wedge F \equiv F$

Commutative Laws: $A \wedge B \equiv B \wedge A$ $A \vee B \equiv B \vee A$

Associative Laws: $(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$ $(A \vee B) \vee C \equiv A \vee (B \vee C)$

Idempotent Laws: $A \wedge A \equiv A$ $A \vee A \equiv A$

Distributive Laws: $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$ $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$

DeMorgan's Laws: $\neg(A \wedge B) \equiv \neg A \vee \neg B$ $\neg(A \vee B) \equiv \neg A \wedge \neg B$

Implication Laws: $A \implies B \equiv \neg B \implies \neg A \equiv \neg A \vee B$

ⓘ Very similar to the algebraic distributive law

TODO: Maybe reformat this as a table so its a bit easier to quickly reference?

Examples

₁ Prove $A \vee (\neg A \wedge B) \equiv A \vee B$

$$\begin{aligned} A \vee (\neg A \wedge B) &\equiv (A \vee \neg A) \wedge (A \vee B) && \text{(Distributive)} \\ &\equiv T \wedge (A \vee B) \\ &\equiv A \vee B && \text{(Identity)} \end{aligned}$$

i In solving these, the goal should be to reduce the # of letters in the propositions. Focus on the side of an equivalence with more going on and try to reduce it down since the more complex proposition will have more opportunities to utilize the different equivalence laws.

₂ Simplify $A \wedge \neg(A \wedge B)$

$$\begin{aligned} A \wedge \neg(A \wedge B) &\equiv A \wedge (\neg A \vee \neg B) && \text{(DeMorgan's)} \\ &\equiv (A \wedge \neg A) \vee (A \wedge \neg B) && \text{(Distributive)} \\ &\equiv F \vee (A \wedge \neg B) \\ &\equiv A \wedge \neg B && \text{(Identity)} \end{aligned}$$

i You don't have to name the laws you're using in the homework, the simple \equiv down the middle format for each step is fine.

₃ Show that $(A \wedge B) \implies (A \vee B)$ is a tautology.

$$\begin{aligned} (A \wedge B) \implies (A \vee B) &\equiv \neg(A \wedge B) \vee (A \vee B) && \text{(Implication)} \\ &\equiv (\neg A \vee \neg B) \vee (A \vee B) && \text{(DeMorgan's)} \\ &\equiv \neg A \vee \neg B \vee A \vee B && \text{(Associative)} \\ &\equiv \neg A \vee A \vee \neg B \vee B && \text{(Commutative)} \\ &\equiv (\neg A \vee A) \vee (\neg B \vee B) && \text{(Associative)} \\ &\equiv T \vee T \\ &\equiv T && \text{(Idempotent)} \end{aligned}$$

1.4 Arguments

 An **argument** is a sequence of propositions in which the conjunction of the initial propositions implies the final proposition

An argument can be represented as;

$$P_1 \wedge P_2 \wedge P_3 \dots \wedge P_n \implies Q$$

Examples

If George Washington was the first president of the United States, then John Adams was the first vice president. George Washington was the first president of the United States. Therefore John Adams was the first vice president.

→ Let A be “George Washington was the first president of the United States.”

→ Let B be "John Adams was the first vice president.”

$$\rightarrow (A \implies B) \wedge A \implies B$$

If Martina is the author of the book, then the book is fiction. But the book is nonfiction. Therefore Martina is not the author.

→ Let A be “Martina is the author of the book.”

→ Let B be “The book is fiction.”

$$\rightarrow (A \implies B) \wedge \neg B \implies \neg A$$


The dog has a shiny coat and loves to bark. Consequently, the dog loves to bark.

→ Let A be “The dog has a shiny coat.”

→ Let B be “The dog loves to bark.”

$$\rightarrow A \wedge B \implies B$$

1.4.1 Valid Arguments / Inference Rules

 An argument is **valid** if and only if **its conclusion is never false while its premises are true**.

We can't use a truth table to validate an argument since it only shows the truth values for the statement as a whole, instead we need to use new **Inference Rules**

Inference Rules

$$\begin{array}{c} P \\ P \implies Q \\ \hline \therefore Q \end{array} \qquad \begin{array}{c} P \implies Q \\ \neg Q \\ \hline \therefore \neg P \end{array}$$

Ex: If George Washington... Ex: If Martina...

$$\begin{array}{c} P \wedge Q \\ \hline \therefore P \end{array} \qquad \begin{array}{c} P \\ \hline \therefore P \vee Q \end{array}$$

$$\begin{array}{c} P \\ Q \\ \hline \therefore P \wedge Q \end{array}$$

Ex: Paul is a good swimmer. Paul is a good runner.

Therefore Paul is a good swimmer and a good runner

i Each line of these rules are basically "if this prop is true and if that prop is true then the last prop is true"

Examples (finding conclusions)

- If the car was involved in the hit-and-run, then the paint would be chipped. But the paint is not chipped.
 - "Car was involved in a hit-and-run" $\rightarrow P$
 - "Paint would be chipped" $\rightarrow Q$
 - "The paint is not chipped" $\rightarrow \neg Q$
 - Conclusion: The car was not involved in a hit-and-run. From the second rule!
- If the bill was sent today, then you will be paid tomorrow. You will be paid tomorrow.
 - Nothing can be concluded from this. ☺
- If the program is efficient_P, it executes quickly_Q. Either the program is efficient_P, or it has a bug_R. However, the program does not execute quickly_{¬Q}.
 - "If the program is efficient" $\rightarrow P$
 - "it executes quickly" $\rightarrow Q$
 - "it has a bug" $\rightarrow R$
 - "the program does not execute quickly" $\rightarrow \neg Q$
 - We start by knowing $P \implies Q$ and $P \vee R$ and $\neg Q$...
 - $(P \implies Q)$ and $\neg Q$ can imply $\neg P$
 - We need to transform $P \vee R$ to use it: $P \vee R \equiv \neg(\neg P) \vee R \equiv \neg P \implies R$
 - $\neg P \implies R$ and $\neg P$ (the first implication we isolated) now implies R by the first inference rule.

1.4.2 Proving a Valid Argument

Assuming the premises are true, apply a sequence of premises and derivation rules, which include the equivalence laws and inference.

General Steps

1. Identify all the premises (might need some transformations).
2. Think backwards. Start from what you want and then seek supporting premises, current results, and necessary equivalence laws and inference rules, until you reach the given premises.
3. Write the proof sequence, where **each step is either one premise or derived from previous step(s) using equivalence laws or inference rules.**

i Start with the RHS of the argument on the bottom of the list and work your way up

Examples

1 Prove $(A \implies B) \wedge (\neg C \vee A) \wedge C \implies B$

i This one is already in its standard form - so we just need to identify each part of the standard $P_1 \wedge P_2 \wedge P_3 \dots \wedge P_n \implies Q$ form. At the end of this we want to prove that B is true.

$$\begin{array}{ccccccc}
 (A \implies B) & \wedge & (\neg C \vee A) & \wedge & C & \implies & B \\
 & \downarrow & \text{Implication Law} & & & & \\
 & \downarrow & \wedge & C \implies A & \wedge & C & \downarrow \\
 (A \implies B) & \wedge & \text{first inference rule law} & A & & & \downarrow \\
 & & \text{also by first inf rule} & B & & \implies & B
 \end{array}$$

i This is not usually how you would format these proofs, this table was to give you an idea of the actual process. The actual proof would look like the following;

1. $A \implies B$
2. $\neg C \vee A$
3. C
4. $C \implies A$ (2, Implication)
5. A (3,4)
6. B (1,5)

You need to put every step in a separate (numbered) line, starting with each component of the argument and then the transformations you do with the reason given. You don't need to name the law used but you need to mention the steps you combined to achieve the next part.

₂ Prove $A \wedge (B \implies C) \wedge ((A \wedge B) \implies (D \vee \neg C)) \wedge B \implies D$

i For this one focus on step 3 ($D \vee \neg C$) as your point to figure out this argument since its the only portion that has D in it.

1. A
2. $B \implies C$
3. $(A \wedge B) \implies (D \vee \neg C)$
4. B
5. $A \wedge B$ (1,4)
6. $D \vee \neg C$ (3,5)
7. $C \implies D$ (6, Commutative, Implication) - Commutative used to swap C, D
8. C (2,4)
9. D (7,8)

TODO: ADD MORE NOTES TO THIS ONE, I GENIUNELY AM NOT FOLLOWING THE PROCESS HERE

₃ Prove $(A \implies B) \wedge (\neg C \vee A) \wedge C \implies A \wedge B$

i For this one notice that the right-hand side isn't a single letter anymore. We now need to focus on proving the whole $A \wedge B$ statement. So this problem is actually solved a bit backwards, start by writing the last steps ($A, B, A \wedge B$) and then go up and figure out how you can prove A .

1. $A \implies B$
2. $\neg C \vee A$
3. C
4. $C \implies A$ (2, Implication)
5. A (3,4)
6. B (1,5)
7. $A \wedge B$ (5,6)

i If instead this problem was looking for $A \vee B$, you could just prove either A or B to make the entire statement valid.

1.4.3 The Deduction Method

Now, what if the conclusion is in implication form?

There are two ways of solving for this form, the main one being **The Deduction Method...**

Suppose the argument has the form:

$$P_1 \wedge P_2 \wedge P_3 \dots \wedge P_n \implies (R \implies S)$$

where the conclusion itself is an implication. We can add R as an additional premise and then imply S.
In other words, we can have the argument:

$$P_1 \wedge P_2 \wedge P_3 \dots \wedge P_n \wedge R \implies S$$

Examples

₁ Prove $(A \implies B) \wedge (B \implies C) \implies (A \implies C)$

❶ *Start with C at the bottom. Now the only way to validate C is if B is true, so make B step 4 and find the proper relations to make B true.*

Deduction: $(A \implies B) \wedge (B \implies C) \wedge A \implies C$

1. $A \implies B$
2. $B \implies C$
3. A
4. B (1,3)
5. C (2,4)

₂ Prove $\neg(A \wedge \neg B) \wedge (B \implies C) \implies (A \implies C)$

Deduction: $\neg(A \wedge \neg B) \wedge (B \implies C) \wedge A \implies C$

1. $\neg(A \wedge \neg B)$
2. $B \implies C$
3. A
4. $\neg A \vee B$ (1, DeMorgan's)
5. $A \implies B$ (4, Implication)
6. B (3,5)
7. C (2,6)

1.5 Predicate Logic

 A **predicate** represents the properties of/relations among objects.

Examples:

n is a perfect square

x is greater than y

Often propositional logic is not enough!

There are several cases where propositional logic won't help us reach needed conclusions or information;

- Suppose we know that "All CS students must take CSCI 358". We cannot conclude that "Alice must take CSCI 358 where Alice is a CS student" using our current propositional logic knowledge.
- Statements that hold many objects must be enumerated;

→ Example:

- * If Alice is a CS student, then Alice must take CSCI358.
- * If Bob is a CS student, then Bob must take CSCI358.
- * If Chris is a CS student, then Chris must take CSCI358.
- * ...

→ Solution: make statements with variables

- * If x is a CS student, then x must take CSCI358.

- Statements that define the property of a group of objects;

→ Example:

- * All new cars must be registered.
- * Some of the new CS students graduate with honor.

→ Solution: Make statements with quantifiers:

- * Universal Quantifier - the property is satisfied by all members of the group.
- * Existential Quantifier - at least one member of the group satisfies the property.

1.5.1 Predicate representation

Predicates are represented like functions in other branches of maths;
e.g $P(x)$ represents some predicate such as "x is a perfect square".

Note that predicates can involve multiple variables, e.g $Q(x,y)$ is "x is greater than y."

❗ Once we plug in a value for x, the predicate becomes a proposition

The two main quantifiers are represented with \forall and \exists

- Universal Quantifier: \forall
 - Read as "for all," "for every," "for each," or "for any."
 - Ex: $\forall x, x > 0$ is read as "for any number x, x is greater than 0."
- Existential Quantifier: \exists
 - Read as "there exists one," "there is," "for at least one," or "for some."
 - Example: $\exists x, x > 0$ is read as "there exists a number x such that x is greater than zero."

❗ When $\forall xP(x)$ or $\exists xP(x)$ is used, the domain must be specified.

Truth Values of Predicates

Predicate	True When...	False When...	Examples
$\forall xP(x)$	If P(x) is true for every x in the domain	If there is any x in the domain such that P(x) is false	$P(x)$ is $x + 1 > x$, $\forall P(x)$ is true for the domain consisting of all real numbers. <hr/> $Q(x)$ is $x < 2$. $\forall xQ(x)$ is false for the domain consisting of all real numbers because $Q(3)$ is false. $x = 3$ is a counterexample of $\forall xQ(x)$
$\exists xP(x)$	There's is an x any-where such that P(x) is true.	P(x) is false for every x	$P(x)$ is $x > 3$. $\exists xP(x)$ is true for the domain consisting of all real numbers. Because when $x=4$, $P(4)$ is true. <hr/> $Q(x)$ is $X = x + 1$. $\exists xQ(x)$ is false for the domain consisting of all real numbers. Because $Q(x)$ is false for every real number x

The quantifiers \forall and \exists have higher precedence than all logical connectives from propositional logic.

For Example:

$\forall xP(x) \wedge Q(x)$ means $(\forall xP(x)) \wedge Q(x)$ rather than $\forall x(P(x) \wedge Q(x))$

Negating Quantified Expressions

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

Example:

→ Every CS Student Must take CSCI385.

→ Negation: There is a CS student who doesn't have to take CSCI358

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

Example:

→ There is s student in this class who has taken CSCI 262.

→ Negation: Every student in this class has **not** taken CS262

1.5.2 Translating to Logical Expressions

Converting statements to expressions

English to Logical Expressions

₁ Every parrot is beautiful

Translation:

→ Assume that the domain consists of all parrots.

* Let $B(x)$ denote "x is beautiful"

* Then $\forall x B(x)$

→ Assume that the domain consists of all animals.

* Let $P(x)$ denote "x is a parrot".

* Let $B(x)$ denote "x is beautiful"

* Then $\forall x (P(x) \implies B(x))$

i
 $\forall x (P(x) \wedge B(x))$
would be incor-
rect

₂ There exists a beautiful parrot

Translation:

→ Assume that the domain consists of all parrots.

* Let $B(x)$ denote "x is beautiful"

* Then $\exists x B(x)$

→ Assume that the domain consists of all animals.

* Let $P(x)$ denote "x is a parrot"

* Let $B(x)$ denote "x is beautiful"

* Then $\exists x (P(x) \wedge B(x))$

i $\exists x (P(x) \implies B(x))$ is an incorrect solution. If x is not a parrot then $P(x)$ is false, since $P(x)$ is attached to the start of the implication it would make the entire expression true (when it should be false)

3 Let $P(x)$ denote "x speaks Russian"
 and let $Q(x)$ denote "x knows the computer language C++"
 Let the domain consist of all students at Mines.
 Translate the following into logical expressions;

There is a student at Mines who speaks Russian and knows C++

$$\rightarrow \exists x (P(x) \wedge Q(x))$$

There is a student at Mines who speaks Russian but doesn't know C++

$$\rightarrow \exists (P(x) \wedge \neg Q(x))$$

Every student at Mines either speaks Russian or knows C++

$$\rightarrow \forall x (P(x) \vee Q(x))$$

No student as Mines speaks Russian or knows C++

$$\rightarrow \forall x (\neg P(x) \wedge \neg Q(x))$$

$$\rightarrow \text{or } \neg \exists x (P(x) \vee Q(x))$$

Nested quantifiers

More than one quantifier may be needed to represent the meaning of a statement in predicate logic.

1 Every real number has its corresponding negative

Assume that the domain consists of all real numbers

Let $P(x, y)$ denote " $x + y = 0$ "

Then we can write $\forall x \exists y P(x, y)$

2 There is a person who loves everybody.

Assume that the domain consists of all people

Let $L(x, y)$ denote " x loves y ".

Then we can write $\exists x \forall y L(x, y)$

Order of quantifiers

When quantifiers are of the **same** type, the order doesn't matter.

Example:

- > Assume that the domain consists of all real numbers.
- > Let $P(x, y)$ denote " $x + y = y + x$ "
- > $\forall x \forall y P(x, y)$ represents "For every real number x , for every real number y , $x + y = y + x$."
- > $\forall y \forall x P(x, y)$ represents "For every real number y , for every real number x , $x + y = y + x$."
- > $\forall x \forall y P(x, y)$ and $\forall y \forall x P(x, y)$ have the same meaning!

When quantifiers are of **different** types, the order does matter!

Example:

- > Assume that the domain consists of all real numbers.
- > Let $Q(x, y)$ denote " $x + y = 0$ "
- > $\forall x \exists y Q(x, y)$ represents "For every real number x , there is a real number y , such that $x + y = 0$."
- > $\exists y \text{ for all } x Q(x, y)$ represents "There is a real number y , such that for every real number x , $x + y = 0$."
- > $\forall x \exists y Q(x, y)$ and $\exists y \text{ for all } x Q(x, y)$ have different meanings!

Ex: Let $Q(x, y, z)$ be " $x + y = z$ " and assume that the domain consists of all real numbers.

$$\forall x \forall y \exists z Q(x, y, z) \not\equiv \exists z \forall x \forall y Q(x, y, z)$$