1. Laws of propositional logic

1.1 Propositions and logical operations

Logic

- Logic is the study of formal reasoning.
- A statement in logic always has a well-defined meaning.
- Logic is important in mathematics for proving theorems.
- Logic is used in computer science for automated reasoning and in designing digital circuits.
- Logic is useful in any field in which it is important to make precise statements.
- In law, logic can be used to define the implications of a particular law.
- In medicine, logic can be used to specify precisely the conditions under which a particular diagnosis would apply.

Propositions

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

Proposition	Truth value
There are an infinite number of prime numbers.	True
17 is an even number.	False

- Propositions are typically declarative sentences. For example, the following are *not* propositions.

Sentence	Comment
What time is it?	A question, not a proposition. A question is neither true nor false.
Are you awake?	Even a yes/no question is neither true nor false, so is not a proposition.
Have a nice day.	A command, not a proposition. A command is neither true nor false.

- A proposition is either true or false, regardless of whether its truth value is known, unknown, or a matter of opinion.

Proposition	Comment
Two plus two is four.	Truth value is true.
Two plus two is five.	Truth value is false.
Monday will be cloudy.	Truth value is unknown.
The movie was funny.	Truth value is a matter of opinion.

The Conjunction Operation

- Propositional variables such as p, q, and r can denote arbitrary propositions.
- A compound proposition is created by connecting individual propositions with logical operations.
- The conjunction operation is denoted by $\land$.
- $p \land q$ is true if both p and q are true.
- $p \land q$ is false if p is false, q is false, or both are false.
- $p \land q$ is read "p and q" and is called the conjunction of p and q.
- The proposition $p \land q$ is expressed in English as: "January has 31 days and February has 33 days."
- A truth table shows the truth value of a compound proposition for every possible combination of truth values for the variables contained in the compound proposition.
- The truth table for $p \land q$, where T represents true and F represents false:

p	q	$p \land q$
T	T	T
T	F	F
F	T	F
F	F	F

Different ways to express a conjunction in English

Define the propositional variables p and h as:

p: The sauce looks disgusting.
h: The sauce tastes delicious.

There are many ways to express the proposition $p \land h$ in English. The sentences below have slightly different meanings in English but correspond to the same logical meaning.

p and h	The sauce looks disgusting and tastes delicious.
p, but h	The sauce looks disgusting, but tastes delicious.
Despite the fact that p, h	Despite the fact that the sauce looks disgusting, it tastes delicious.
Although p, h	Although the sauce looks disgusting, it tastes delicious.
The movie was funny.	Truth value is a matter of opinion.

The Disjunction Operation

- The disjunction operation is denoted by $\lor$.
- $p \lor q$ is read "p or q" and is called the disjunction of p and q.
- $p \lor q$ is true if either one of p or q is true, or if both are true.
- The proposition $p \lor q$ is false only if both p and q are false.
- The proposition $p \lor q$ is expressed in English as: "January has 31 days or February has 33 days."
- The truth table for $p \lor q$, where T represents true and F represents false:

p	q	$p \lor q$
T	T	T
T	F	T
F	T	T
F	F	F

Ambiguity of "or" in English

- The meaning of the word "or" in common English depends on context.
- Often, "or" in English means that one or the other of two things is true, but not both, which corresponds to the "exclusive or" operation in logic.

- The exclusive or of p and q evaluates to true when p is true and q is false or when q is true and p is false.
- The inclusive or operation is the same as the disjunction (∨) operation and evaluates to true when one or both of the propositions are true.
- In logic, the inclusive or is more commonly used and is just called "or" for short.
- For example, the sentence "Lucy is going to the park or the movie" is understood to mean that Lucy is either going to the park or the movie, but not both. However, the sentence "Lucy opens the windows or doors when warm" means she opens windows, doors, or possibly both.

1.2 Evaluating compound propositions

Compound Propositions and Order of Operations

- Compound propositions can use more than one operation.
- The order of operations for compound propositions without parentheses is ¬, ∧, ∨.
- It is good practice to use parentheses to specify the order of operations.
- Parentheses around ¬p are usually omitted for readability.
- Multiple ∨ or ∧ operations usually do not need parentheses.
- Exclusive or (⊕) is discussed later.

Order of operations in absence of parentheses

Priority	Operation
1	¬ (not)
2	∧ (and)
3	∨ (or)

Evaluate Expression

$p \land \neg (q \lor r)$

When:
p = T
q = F
r = T

Expression	
p ∧ ¬ (q ∨ r)	Fill in truth values
T ∧ ¬(F ∨ T)	Evaluate operations in order of priority
T ∧ ¬T	Parenthesis
T ∧ F	Not Operation
F	And Operation

Filling in Truth Tables

- A truth table for a compound proposition has a row for every possible combination of truth assignments for the statement's variables.
- If a compound proposition has n variables, there are 2^n rows.
- Each column is labeled, with variable columns on the left and the compound proposition column on the right.
- To fill in the variable columns, start with the right-most column and fill in an alternating T and F pattern from top to bottom.
- Each subsequent column is filled in with a pattern that doubles the number of T's and F's from the previous column.

p	q	r	((p ∨ r) ∧ ¬q)
T	T	T	F
T	T	F	F
T	F	T	T
T	F	F	T
F	T	T	F
F	T	F	F
F	F	T	T
F	F	F	F

Example: Electronic Fan Control Propositions (example)

Everyday devices operate through electronic circuitry that follows logical laws. To ensure desired behavior, a designer must express the logic and test it under various conditions.

An electronic fan turns on/off based on room humidity. To compensate for an imprecise humidity detector, the fan runs for 20 mins to clear moisture. A manual switch can override automatic control.

The operation of this fan can be modeled as an expression:

- M: The fan has been on for twenty minutes.
- H: The humidity level in the room is low.
- O: The manual "off" button has been pushed.
- $(M \wedge H) \vee O$

A truth table tests a device to ensure proper operation under all conditions. A technician might use this table to test the electronic fan.

M	H	O	Should be off? (T: yes)
T	T	T	T
T	T	F	T
T	F	T	T
T	F	F	F
F	T	T	T
F	T	F	F
F	F	T	T
F	F	F	F

1.3 Conditional statements

Conditional Propositions (Implication)

- A conditional proposition uses the symbol → and is read "if p then q"
- p is the hypothesis and q is the conclusion
- A conditional proposition can be compared to a contract where the only way for it to be false is if the hypothesis is true and the conclusion is false.
- The truth table for $p \rightarrow q$ is:

p	q	p → q
T	T	T
T	F	F
F	T	T
F	F	T

"If you water the plants regularly, then they will grow healthy and strong."

P: you water the plants regularly
Q: the plants grow healthy and strong

P → Q

Alternative English expressions of the condtional operation

Statement	Meaning
If p, then q	If you water the plants regularly, then they will grow healthy and strong.
If p, q	If you water the plants regularly, they will grow healthy and strong.
q if p	The plants will grow healthy and strong if you water them regularly.
p implies q	Watering the plants regularly implies that they will grow healthy and strong.
p only if q	You will water the plants regularly only if they will grow healthy and strong.
p is sufficient for q	Watering the plants regularly is sufficient for them to grow healthy and strong.
q is necessary for p	The plants growing healthy and strong is necessary for you to water them regularly.

- Do not confuse "only if" (Conditional) with "if and only if" (Bi-Conditional).

Converse, Contrapositive, and Inverse

- Three conditional statements related to p → q:
  - Converse: q → p
  - Contrapositive: ¬q → ¬p
  - Inverse: ¬p → ¬q
- The converse switches the hypothesis and conclusion of the conditional statement
- The contrapositive negates both the hypothesis and conclusion of the conditional statement and switches their order
- The inverse negates both the hypothesis and conclusion of the conditional statement.

Statement	Logical Notation	Meaning
Proposition:	p → q	If it rains, then the ground gets wet.
Converse:	q → p	If the ground gets wet, then it rains.
Contrapositive:	¬q → ¬p	If the ground does not get wet, then it is not raining.
Inverse:	¬p → ¬q	If it does not rain, then the ground does not get wet.

Biconditional Operation

- The biconditional operation is denoted with ↔ and means "p if and only if q"
- p ↔ q is true if p and q have the same truth value; otherwise, it is false
- Alternative ways of expressing p ↔ q include "p is necessary and sufficient for q" or "if p then q, and conversely"
- The truth table for p ↔ q is:

p	q	p ↔ q
T	T	T
T	F	F
F	T	F
F	F	T

Compound propositions with conditional and biconditional operations

- The conditional and biconditional operations can be combined with other logical operations, as in (p → q) ∧ r.
- If parentheses are not used to explicitly indicate the order in which the operations should be applied, then ∧, ∨, and ¬ should be applied before → or ↔.
- Thus, the proposition p → q ∧ r should be evaluated as p → (q ∧ r).
- Good practice, however, is to use parentheses so that the order of operations is clear.

Example: Requirement checks

Large universities often use automated systems to check if a student has fulfilled the degree requirements before graduation. Requirements can be expressed in logic, such as (P1 ∨ P2 ∨ P3) ∧ (T1 ∨ T2) ∧ (H → S) for a Computer Science degree where students must take one project course and one theory course, and an honors seminar if they are an honors student.

1.4 Logical equivalence

Tautology and Contradiction

- A compound proposition is a tautology if it's always true regardless of the truth value of the individual propositions that occur in it
- A compound proposition is a contradiction if it's always false regardless of the truth value of the individual propositions that occur in it
- Example of a tautology is p ∨ ¬p, as it's always true

p	¬p	p ∨ ¬p
T	F	T
F	T	T

- Example of a contradiction is p ∧ ¬p, as it's always false

p	¬p	p ∧ ¬p
T	F	F

p	¬p	p ∧ ¬p
F	T	F

- Truth tables can be used to verify if a compound proposition is a tautology or contradiction
- To show that a compound proposition is not a tautology, a set of truth values causing the compound proposition to be false needs to be shown
- To show that a compound proposition is not a contradiction, a set of truth values causing the compound proposition to be true needs to be shown

Showing Logical Equivalence Using Truth Tables

- Two compound propositions are logically equivalent if they have the same truth value regardless of the truth values of their individual propositions
- The notation s ≡ r is used to indicate that r and s are logically equivalent
- Propositions s and r are logically equivalent if and only if the proposition s ↔ r is a tautology
- To show that two compound propositions are logically equivalent, a truth table can be used
- Note that s ≡ r if and only if r ≡ s

Contrapositive truth table: (p → q) ≡ ¬ q → ¬ p

p	q	(p → q)	¬q → ¬p
T	T	T	T
T	F	F	F
F	T	T	T
F	F	T	T

To show that two propositions are not logically equivalent, demonstrate a set of truth values for their individual propositions that cause the two compound propositions to have different truth values. For instance, p ↔ r and p → r are not equivalent when p = F and r = T because p ↔ r is false, while p → r is true.

De Morgan's Laws

- De Morgan's laws are logical equivalences that show how to correctly distribute a negation operation inside a parenthesized expression
- The first De Morgan's law is: $\neg(p \lor q) \equiv (\neg p \land \neg q)$
- When the negation operation is distributed inside the parentheses, the disjunction operation changes to a conjunction operation.
- The second De Morgan's laws is: $\neg(p \land q) \equiv (\neg p \lor \neg q)$
- The second version of De Morgan's law swaps the role of the disjunction and conjunction.
- De Morgan's laws are particularly useful in logical reasoning

De Morgan's law applied to an English statement:

"Not going to the beach or the park" is equivalent to "Going neither to the beach nor the park".

Truth tables

Disjunctive

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

Conjunctive

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

1.5 Laws of propositional logic

Logical equivalence

Logical equivalence allows for the substitution of equivalent propositions within a more complex proposition, resulting in a logically equivalent compound proposition.

$p \to q \equiv \neg p \lor q$ is the conditional equivalence, and it (along with other equivalences) allow expressions to be converted between forms that might be easier to work with.

For example, $(p \lor r) \land (\neg p \lor q)$ can be converted to: $(p \lor r) \land (p \to q)$.

Conditional equivalence truth table

p	q	p → q	(¬p ∨ q)
T	T	T	T
T	F	F	F
F	T	T	T
F	F	T	T

Laws of propositional logic

To show logical equivalence between two propositions, substitution can be used. If equivalent expressions can be substituted in one proposition to obtain the other, the two propositions are logically equivalent. The table below displays some useful laws of propositional logic for establishing equivalence between compound propositions.

Idempotent laws:	$p \lor p \equiv p$	$p \land p \equiv p$
Associative laws:	$(p \lor q) \lor r \equiv p \lor (q \lor r)$	$(p \land q) \land r \equiv p \land (q \land r)$
Commutative laws:	$p \lor q \equiv q \lor p$	$p \land q \equiv q \land p$
Distributive laws:	$p \lor (q \land r) \equiv (p \lor q) \land (p \lor r)$	$p \land (q \lor r) \equiv (p \land q) \lor (p \land r)$
Identity laws:	$p \lor F \equiv p$	$p \land T \equiv p$
Domination laws:	$p \land F \equiv F$	$p \lor T \equiv T$
Double negation law:	$\neg\neg p \equiv p$	
Complement laws:	$p \land \neg p \equiv F$ $\neg T \equiv F$	$p \lor \neg p \equiv T$ $\neg F \equiv T$
De Morgan's laws:	$\neg(p \lor q) \equiv \neg p \land \neg q$	$\neg(p \land q) \equiv \neg p \lor \neg q$
Absorption laws:	$p \lor (p \land q) \equiv p$	$p \land (p \lor q) \equiv p$

Idempotent laws:	p ∨ p ≡ p	p ∧ p ≡ p
Conditional identities:	p → q ≡ ¬p ∨ q	p ↔ q ≡ (p → q) ∧ (q → p)

1.6 Predicates and quantifiers

Mathematical Statements with Variables

- Many mathematical statements contain variables
- The statement "x is an odd number" is not a proposition
- It does not have a well-defined truth value until the value of x is specified
- The truth value of the statement can be expressed as a function P of the variable x, as in P(x)
- A logical statement whose truth value is a function of one or more variables is called a predicate
- If P(x) is defined to be the statement "x is an odd number", then P(5) corresponds to the statement "5 is an odd number"
- P(5) is a proposition because it has a well-defined truth value.
- A predicate can depend on more than one variable.
- Define the predicates Q and R as:

$$Q(x, y) : x^2 = y$$
$$R(x, y, z) : x + y = z$$

- The proposition Q(5, 25) is true because $5^2 = 25$.
- The proposition R(2, 3, 6) is false because 2 + 3 ≠ 6.

Domains in Predicates

- The domain of a variable in a predicate is the set of all possible values for the variable
- A natural domain for the variable x in the predicate "x is an odd number" would be the set of all integers
- If the domain of a variable in a predicate is not clear from context, the domain should be given as part of the definition of the predicate.

Predicates in Non-Mathematical Statements

- Statements outside the realm of mathematics can also be predicates
- For example, consider the statement: "The city has a population over 1,000,000."
- The "city" is the variable and the domain is defined to be the set of all cities in the United States
- When the city is New York, the statement becomes: "New York has a population over 1,000,000" and the statement is true. When the city is Toledo, the statement becomes: "Toledo has a population over 1,000,000" and the statement is false
- A statement P(x) may be true for all values in the domain, but if it contains a variable, it is still considered to be a predicate and not a proposition
- For example, if P(x) is the statement "x + 1 > 1" and the domain is all positive integers, the statement is true for each value in the domain, but P(x) is considered to be a predicate and not a proposition because it contains a variable.

Universal Quantifier

- Another way to turn a predicate into a proposition is to use a quantifier
- The logical statement ∀x P(x) is read "for all x, P(x)" or "for every x, P(x)"
- The statement ∀x P(x) asserts that P(x) is true for every possible value for x in its domain
- The symbol ∀ is a universal quantifier and the statement ∀x P(x) is called a universally quantified statement
- ∀x P(x) is a proposition because it is either true or false
- ∀x P(x) is true if and only if P(n) is true for every n in the domain of variable x.

If the domain is a finite set of elements {a_1, a_2, ..., a_k}, then:

$$\forall x\ P(x)\ \equiv\ P(a_1)\ \land\ P(a_2)\ \land\ \cdots\ \land\ P(a_k)$$

Equivalence Symbol

- The equivalence symbol means that the two expressions always have the same truth value, regardless of the truth values for P(a_1) to P(a_n).
- If the domain is the set of students in a class and the predicate A(x) means that student x completed the assignment, then the proposition ∀x A(x) means: "Every student completed the assignment."

- Establishing that ∀x A(x) is true requires showing that each and every student in the class did in fact complete the assignment.

Arbitrary Element

- Some universally quantified statements can be shown to be true by showing that the predicate holds for an arbitrary element from the domain.
- An "arbitrary element" means nothing is assumed about the element other than the fact that it is in the domain.
- In the following example, the domain is the set of all positive integers:

$$\forall x(\frac{1}{x+1} < 1)$$

- The statement is true because when x is assigned any arbitrary value from the set of all positive integers, the inequality 1/(x+1)<1 holds.

Counterexamples in Universal Quantification

- A counterexample for a universally quantified statement is an element in the domain for which the predicate is false.
- A single counterexample is sufficient to show that a universally quantified statement is false.
- Example:
  ∀x (x^2 > x) with domain set of positive integers
  x = 1 is a counterexample since x^2 = x and the statement 1 > 1 is not true.

Existential Quantifier

- The logical statement ∃x P(x) is read "There exists an x, such that P(x)".
- ∃x P(x) asserts that P(x) is true for at least one possible value for x in its domain.
- ∃ is an existential quantifier and ∃x P(x) is called an existentially quantified statement.
- ∃x P(x) is a proposition because it is either true or false.
- ∃x P(x) is true if and only if P(n) is true for at least one value n in the domain of variable x.

If the domain is a finite set of elements {a_1, a_2, ..., a_k}, then:

$$\exists x\ P(x)\ \equiv\ P(a_1)\ \vee\ P(a_2)\ \vee\ \ldots\ \vee\ P(a_k)$$

Existential Quantified Statement with Predicate

- If the domain is the set of students in a class and the predicate A(x) means that student x completed the assignment, then ∃x A(x) is the statement: "There is a student who completed the assignment."
- To establish that ∃x A(x) is true, it only requires finding one particular student who completed the assignment.
- An example for an existentially quantified statement is an element in the domain for which the predicate is true.
- A single example is sufficient to show that an existentially quantified statement is true.
- To show that ∃x A(x) is false, it requires showing that every student in the class did not complete the assignment.
- Some existentially quantified statements can be shown to be false by showing that the predicate is false for an arbitrary element from the domain.

For example, consider the existentially quantified statement in which the domain of x is the set of all positive integers:

$$\exists x(x+1<x)$$

The statement is false because no positive integer satisfies the expression x + 1 < x.

1.7 Quantified statements

- Note: the Uniqueness Quantifier is not mentioned here.

Quantified Statements with Predicates and Logical Operations

- Universally and existentially quantified statements can be constructed from logical operations.
- For example, consider the domain of positive integers and the predicates P(x): x is prime and O(x): x is odd.
- The proposition ∃x (P(x) ∧ ¬O(x)) states that there exists a positive number that is prime and not odd.
- This proposition is true because of the example x = 2.
- The proposition ∀x (P(x) → O(x)) says that for every positive integer x, if x is prime then x is odd.

- This proposition is false because of the counterexample x = 2.
- The universal and existential quantifiers are generically called quantifiers, and a logical statement that includes a universal or existential quantifier is called a quantified statement.
- The quantifiers ∀ and ∃ are applied before the logical operations (∧, ∨, →, and ↔) used for propositions.

Free and Bound Variables

- A variable x in the predicate P(x) is a free variable because it can take on any value in the domain.
- A variable x in the statement ∀x P(x) is a bound variable because it is bound to a quantifier.
- A statement with no free variables is a proposition because its truth value can be determined.

Bound and Free Variables in Quantified Statements

- In the statement (∀x P(x)) ∧ Q(x), the variable x in P(x) is bound by the universal quantifier, but the variable x in Q(x) is a free variable and is not bound by the universal quantifier.
- Therefore, the statement (∀x P(x)) ∧ Q(x) is not a proposition.
- In contrast, the universal quantifier in the statement ∀x (P(x) ∧ Q(x)) binds both occurrences of the variable x.
- Therefore, ∀x (P(x) ∧ Q(x)) is a proposition.

Logical equivalence with quantified statements

- Two quantified statements have the same logical meaning if they have the same truth value regardless of the value of the predicates for the elements in the domain.

Example: consider a domain consisting of a set of people invited to go fishing.

- P(x): x went fishing
- S(x): x was sick

Person	S(x)	P(x)	¬S(x) ↔ P(x)

Person	S(x)	P(x)	$\neg S(x) \leftrightarrow P(x)$
Alice	F	T	T
Bob	F	T	T
Charlie	T	F	T
David	F	T	T
Emily	T	F	T

- The statement "everyone went fishing" is logically equivalent to "$\forall x\ P(x)$"
- The statement "everyone was not sick" is logically equivalent to "$\forall x\ \neg S(x)$"
- So, "everyone went fishing if and only if they were not sick" is logically equivalent to "$\forall x\ (\neg S(x) \leftrightarrow P(x))$"
- $\forall x\ (\neg S(x) \leftrightarrow P(x))$ = True because everyone who went fishing was not sick, and everyone who was not sick went fishing.

Example: Translating quantified statements from English to logic

Consider the following problem where the domain is a group of students in a classroom and the predicates are:

- H(x): person x has a laptop.
- U(x): person x uses the laptop during class.

Translate each of the following sentences into an equivalent logical expression:

1. All students who have a laptop use it during class.

   $\forall x\ (H(x) \rightarrow U(x))$.

   For every student x, if they have a laptop (H(x)) then they use it during class (U(x)).

2. There exists a student who has a laptop but does not use it during class.

   $\exists x\ (H(x) \land \neg U(x))$.

   There exists a student x, such that they have a laptop (H(x)) but do not use it during class ($\neg$U(x)).

3. No student uses a laptop during class.

   ∀x (¬U(x)).

   For every student x, they do not use a laptop during class (¬U(x)).

1.8 De Morgan's law for quantified statements

- De Morgan's law for quantified statements is formally stated as ¬∀x F(x) ≡ ∃x ¬F(x)
- For a finite domain, De Morgan's law for universally quantified statements is the same as De Morgan's law for propositions.

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

Example: flappy birds

Let the domain of x be the set of all birds.
F(x): x can fly

The statement:
"Not every bird can fly" can be expressed as: ¬∀x F(x)

Which is equivalent to:
"There exists a bird that cannot fly", which can be expressed as ∃x ¬F(x)

Example: imposters

Let the domain x be the set of all crew members on a space station.
H(x): x is human
B(x): x is breathing

The statement:
"No human on this space station is breathing" can be expressed as ¬∃x (H(x) ∧ B(x))

Which is equivalent to:
"All crew members on the space station are not"

Example: simplify an existentially quantified statement with De Morgan's law

Existential Statement
"No one who loves Star Wars can resist buying Baby Yoda toys."

Where
P(x): x loves Star Wars
Q(x): x can resist buying Baby Yoda toys

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

Expression	Operation
$\neg\exists$ x (P(x) $\wedge$ Q(x))	
$\forall$x $\neg$(P(x) $\wedge$ Q(x)	De Morgan's Law
$\forall$x ($\neg$P(x) $\vee$ $\neg$Q(x))	De Morgan's Law
$\forall$x (P(x) $\rightarrow$ $\neg$Q(x))	Conditional Identity

Universal Statement
"Everyone who loves Star Wars can't help but buy Baby Yoda toys. "

1.9 Nested quantifiers

- If a predicate has more than one variable, each variable must be bound by a separate quantifier.
- A logical expression with more than one quantifier that binds different variables in the same predicate is said to have nested quantifiers.
- The logical expression is a proposition if all the variables are bound.

Examples

Expression	Quantifiers Binding Variables
$\forall$x P(x, y)	x is bound by a universal quantifier and y is free

Expression	Quantifiers Binding Variables
$\forall x \exists y\ P(x, y)$	x is bound by a universal quantifier and y is bound by an existential quantifier.
$\forall x \forall y\ P(x, y)$	x and y are both bound by universal quantifiers.
$\exists x \forall y\ P(x, y)$	x is bound by an existential quantifier and y is bound by a universal quantifier.
$\exists x\ P(x) \land \forall y\ Q(x, y)$	x is bound by an existential quantifier in the left expression and by a universal quantifier in the right expression. y is bound by a universal quantifier in the right expression.

Nested quantifiers of the same type

Suppose there is a group of people working on a project. Let M(x, y) be the predicate that person x sent an email to person y.

If we express the proposition $\forall x\ \forall y\ M(x, y)$ in English, it means "Everyone sent an email to everyone." This statement is true if for all pairs of people x and y, M(x, y) is true, including when x = y. Even if there is a single person who did not send an email to themselves, the statement $\forall x\ \forall y\ M(x, y)$ is still false if there is any pair (x, y) that causes M(x, y) to be false.

On the other hand, if we express the proposition $\exists x\ \exists y\ M(x, y)$ in English, it means "There is a person who sent an email to someone." This statement is true if there exists a pair of people (x, y) for which M(x, y) is true. Even if a single person sent an email to themselves, the statement $\exists x\ \exists y\ M(x, y)$ is still true. The statement $\exists x\ \exists y\ M(x, y)$ is false only when all pairs (x, y) cause M(x, y) to be false.

Domain
x and y is a group of people working at a company

Definition
M(x, y): x sent an email to y

"Everyone sent an email to everyone."
- $\forall x\ \forall y\ M(x, y)$
- True if M(x, y) is true for every pair of x and y, including x=y
- False if M(x, y) is false for any pair of x and y

"There is a person who sent an email to someone."
- ∃x ∃y M(x, y)
- True if there exists at least one pair of x and y that causes M(x, y) to be true
- False if M(x, y) is false for all pairs of x and y

Alternating Nested Quantifiers

- Quantified expressions can contain both types of quantifiers
  ∃x ∀y M(x, y) ↔ "There is a person who sent an email to everyone."
- Switching the order of the quantifiers changes the statement's truth value and meaning
  ∀x ∃y M(x, y) ↔ "Every person sent an email to someone."
- Nested quantifiers are like a two-player game in which two players compete to set the statement's truth value

Player	Action	Goal
Existential player	Selects values for existentially bound variables	Tries to make the expression true
Universal player	Selects values for universally bound variables	Tries to make the expression false

- If predicate is true after all variables are set, then quantified statement is true
- If predicate is false after all variables are set, then quantified statement is false

De Morgan's law with nested quantifiers

- Each time the negation sign moves past a quantifier, the quantifier changes type from universal to existential or from existential to universal.

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

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

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

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

Example: crushes

Domain
x and y are the set of all students in a school

Definition
L(x, y): x likes y

Statement
"There is a student who likes everyone in the school"
$\exists x \forall y L(x, y)$

Negated
"There is no student who likes everyone in the school."
$\neg\exists x \forall y L(x, y)$

Transformed
"Every student in the school has someone that they do not like."
$\forall x \exists y \neg L(x, y)$

1.10 More nested quantified statements

Logic and Email Correspondence

In a group project scenario, we can use logic to talk about email correspondence. To express that everyone sent an email to everyone else, we can use the conditional operation $(x \neq y) \rightarrow M(x, y)$. For example, in a group of four people, the table below shows who sent emails to whom.

Predicate
M(x, y): x sent an email to y

Statement
"Every person sent an email to every other person and every person sent an email to himself or herself."
$\forall x \forall y M(x, y)$

If no one sent an email to themselves, then the truth table would be:

	Agnes	Fred	Sue	Marge
Agnes	F	T	T	T

	Agnes	**Fred**	**Sue**	**Marge**
Fred	T	F	T	T
Sue	T	T	F	T
Marge	T	T	T	F

Problem
Under these conditions, $\forall x \, \forall y \, M(x, y)$ evaluates to false.

Solution
To exclude the case where every person must send an email to themselves, we can include a conditional operation to assert "everyone else"

Updated Statement
"Every person sent an email to every other person."
$\forall x \, \forall y \, ((x \neq y) \rightarrow M(x, y))$

Expressing uniqueness in quantified statements

Existentially quantified statements evaluate to true if there is at least one element in the domain that satisfies the predicate, even if there is more than one.

Example

Definition
$L(x)$: x came late to work

Statement
"Someone came late to work."
$\exists x \, L(x)$

Problem
How can we show that exactly one person came late to the meeting?

Solution
Show that x was late and that no one else was late using a conjunction

Updated Statement
"Exactly one person came late to work."
$\exists x(L(x) \land \forall y((x \neq y) \rightarrow \neg L(x)))$

Moving quantifiers in logical statements

Domain
The set of people on a deserted island.

Definitions
$F(x, y)$: x is bunking with y
$H(x)$: x is happy

Statement
"Every happy person is bunking with someone else."
$\forall x \, (H(x) \rightarrow \exists y \, F(x, y))$

Note
- Since y does not appear in the predicate $H(x)$, "$\exists y$" can be moved to the left so that it appears just after $\forall x$
- However, a quantifier can not be moved in front of another quantifier without changing the meaning of the expression
- For example, $\forall x \, \exists y (H(x) \rightarrow F(x, y))$ is not logically equivalent to $\exists y \, \forall x (H(x) \rightarrow F(x, y))$.

Updated Statement
$\forall x \, \exists y (H(x) \rightarrow F(x, y))$

Example: every happy person is bunking with exactly one person

Start with "every happy person is bunking with someone else"
$\forall x \, (H(x) \rightarrow \exists y \, F(x, y))$

Add the condition that "everyone else is not bunking with x"
$\forall x (H(x) \rightarrow (\exists y \, F(x, y) \land \forall z ((z \neq y) \rightarrow \neg F(x, z))))$

1.11 Logical reasoning

Logic and Arguments

- Language of logic allows formal establishment of truth of logical statements
- An argument is a sequence of hypotheses followed by a conclusion
- An argument is valid if conclusion is true whenever all hypotheses are true, otherwise it's invalid

Notation:

$$\begin{array}{l} p1 \\ p2 \\ .... \\ pn \\ \hline \therefore c \end{array}$$

- Validity: ($p_1 \land p_2 \land \ldots \land p_n$) → c is a tautology
- The commutative law allows reordering of hypotheses without changing validity
- Two arguments are considered the same if hypotheses appear in a different order

The following two arguments are the same:

$$\begin{array}{l} p \\ p \to q \\ \hline \therefore q \end{array} \qquad \begin{array}{l} p \to q \\ p \\ \hline \therefore q \end{array}$$

Validating Arguments with Truth Tables

To use a truth table for validating an argument, we need to construct a table for all the hypotheses and the conclusion. Then, examine each row where all hypotheses are true. If the conclusion is true for all these rows, the argument is valid. But, if there exists a row where all hypotheses are true and the conclusion is false, then the argument is invalid.

Consider the argument:

$$\begin{array}{l} p \to q \\ p \lor q \\ \hline \therefore q \end{array}$$

p	q	p → q	p ∨ q
T	T	T	T

p	q	p → q	p ∨ q
T	F	F	T
F	T	T	T
F	F	T	F