Logic

Propositions and logical operations

Main concepts:

- **propositions**
- **truth values**
- propositional **variables**
- logical **operations**

Propositions and logical operations

- A **proposition** is the most basic element of logic
- It is a declarative sentence that is either **true** or **false**

Propositions and logical operations

Examples of propositions:

- Grass is green.
- The Moon is made of green cheese.
- Sacramento is the capital of California.
- $1 + 0 = 1$
- $0 + 0 = 2$

Propositions and logical operations

Examples that are **not** propositions

- Sit down!
- What time is it?
- $x + 1 = 2$
- $x + y > z$

Propositions and logical operations

- Every proposition has a **truth value (T or F)**
- The value may be:
  - known/widely accepted as **true**
  - known/widely accepted as **false**
  - **unknown**
  - a matter of **opinion** (true for some people)
  - or even a **false belief**

Propositions and logical operations

- All these **are** propositions:

Proposition	Truth value
1 + 1 = 2	True
1 + 1 = 1	False
It will rain tomorrow	Unknown
Logic is boring	Opinion
The sun orbits around the earth	False belief

Constructing Propositions

- To avoid writing long propositions we use propositional **variables**
- A propositional variable is typically a single letter (p, q, r, …)
- It can denote arbitrary propositions
- **Examples**:

  p: it is raining

    p represents the proposition "it is raining"

  q: the streets are wet

    q represents the proposition "the streets are wet"

Compound Propositions

- A **logical operation** combines propositions using certain rules
- **Example**:
  - The operation denoted by "∧" means "**and**"
  - "p ∧ q" means "it is raining **and** the streets are wet"
  - If both p and q are true then p ∧ q is true
  - If either p or q (or both) are false then p ∧ q is false
- "∧" is called the **conjunction**

Constructing Propositions

- **Operations** to construct compound propositions:
  - Conjunction ∧ AND
  - Disjunction ∨ OR
  - Negation ¬ NOT
  - Implication → IF-THEN
  - Biconditional ↔ IFF

Truth Tables

- Any proposition can be represented by a **truth table**
- It shows truth values for **all combinations** of its constituent variables
- **Example**: proposition r involving 2 variables p and q

all possible **combinations** of truth values of p and q		truth values of compound proposition r
p	**q**	**r**
true	true	
true	false	
false	true	
false	false	

Conjunction

- The **conjunction** of propositions p and q is denoted by p ∧ q
- Its truth table is:

p	q	p ∧ q
T	T	T
T	F	F
F	T	F
F	F	F

Example:

- p: I am at home
- q: It is raining
- p ∧ q: I am at home **and** it is raining

Disjunction

- The **disjunction** of propositions p and q is denoted by p ∨ q
- Its truth table is:

p	q	p ∨ q
T	T	T
T	F	T
F	T	T
F	F	F

Example:

- p: I am at home
- q: It is raining
- p ∨ q: I am at home or it is raining

Ambiguity of "or" in English

- In natural languages "or" has two distinct meanings
- **Inclusive Or**:
  - p ∨ q is true if either p or q **or both** are true
- **Example**:
  - "Math 10a or Math 12 may be taken as a prerequisite for CS 6"
  - Meaning: take either one but may also take both

Ambiguity of "or" in English

- **Exclusive Or** (Xor)
  - Denoted as "⊕"
  - p ⊕ q is true if either p or q but **not both** are true

p	q	p ⊕ q
T	T	F
T	F	T
F	T	T
F	F	F

Example:
- "Soup or salad comes with this entrée"
- Meaning: do not expect to get **both**

Negation

- The **negation** of a proposition p is denoted by ¬p
- Its truth table is:

p	¬p
T	F
F	T

Example:
- p: The earth is round
- ¬p: It is **not true** that the earth is round

  or more simply: The earth is **not** round

Constructing truth tables

- A proposition can involve any number of variables
- Each row corresponds to a possible combination of variables
- With n variables the truth table has:
  - **n+1 columns** (1 for each of the n variables and 1 for the compound expression)
  - **2^n rows** (plus a header)

Constructing truth tables

- To construct the truth table for a given proposition:
  1. Create a table with 2^n rows and n+1 columns
  2. Fill in the first n columns with all possible combinations
  3. Determine and enter the truth value for each combination

Constructing truth tables

- **Example**:
  ¬p ∨ (q ∧ r)

p	q	r	¬p ∨ (q ∧ r)

Constructing truth tables

- Using intermediate columns

p	q	r			¬p ∨ (q ∧ 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			

Conditional statements

- If p and q are propositions then

  p → q is a **conditional statement** or **implication**
- It means: if p then q
- p is the **hypothesis** (premise) and q is the **conclusion** (consequence)
- **Example**:
  - p: It is raining
  - q: The streets are wet
  - p → q: **If** it is raining **then** streets are wet
- **From a false premise anything can be implied!**

p	q	p → q
T	T	T
T	F	F
F	T	T
F	F	T

Understanding Implication

- View logical conditional as an **obligation** or **contract**:
- **Example**: "If you get 100% on the final then you will earn an A"

  p → q

	q: earn A	¬q: don't get A
p: get 100%	**p → q = T**	**p → q = F**
¬p: don't get 100%	**p → q = T**	**p → q = T**

- F → T does **not** violate the contract
- The **only** time the contract is broken is when T → F

Understanding Implication

- In p → q there may not be any **connection** between p and q
- **Examples** of **valid** but **counterintuitive** implications:
  - If the moon is made of green cheese then you get a PhD in physics
    - **True!**
  - If Juan has a smartphone then 2 + 3 = 6:
    - **False** if Juan **does** have a smartphone
    - **True** if he **does NOT**

Different Ways of Expressing p → q

if p **then** q	If it's raining then streets are wet
if p, q	If it's raining, streets are wet
p **implies** q	Rain implies that streets are wet
p **only if** q	
q **if** p	
q **when** p	Streets are wet when it's raining
q **whenever** p	Streets are wet whenever it's raining
q **follows from** p	Streets being wet follows from there being a rain
p **is sufficient for** q	
q **is necessary for** p	

Different Ways of Expressing p → q

p → q: It is raining → streets are wet

Which statements are equivalent?

q **if** p	Streets are wet **if** it's raining = T
q **only if** p	Streets are wet **only if** it's raining = F
p **if** q	It's raining **if** streets are wet = F
p **only if** q	It's raining **only if** streets are wet = T

Last statement is awkward but consider a different example:

(n is even → n+1 is odd) ≡ (n is even **only if** n+1 is odd)

Remember: (q if p) ≡ (p only if q) ≢ (p if q) ≡ (q only if p)

Sufficient versus Necessary

p **is sufficient for** q	Rain is **sufficient** for streets being wet
q **is necessary for p**	Wet streets are **necessary** for there being rain

- "Necessary condition" is counter-intuitive in English
- It suggest that wet streets are a requirement for rain
- Better: implicitly read "necessary result of" or "necessary consequence of"

Converse, Inverse, Contrapositive

- From $p \rightarrow q$ we can form new conditional statements:

$q \rightarrow p$	is the **converse** of $p \rightarrow q$
$\neg p \rightarrow \neg q$	is the **inverse** of $p \rightarrow q$
$\neg q \rightarrow \neg p$	is the **contrapositive** of $p \rightarrow q$

- How are these statements related to the original?
- How are they related to each other?
- Are any of them equivalent?

Converse, Inverse, Contrapositive

- **Example**
  1. **implication:** it's raining → streets are wet
     True
  2. **converse**: streets are wet → it's raining
     False
  3. **inverse**: it's **not** raining → streets are **not** wet
     False
  4. **contrapositive**: streets are **not** wet → it's **not** raining
     True

  Only **1 = 4** and **2 = 3**

Biconditional

- The **biconditional** proposition p ↔ q means: **p if and only if** q
- Other ways to say this:
  - p **iff** q
  - **if** p **then** q, **and conversely**
  - p **is necessary and sufficient for** q

p	q	p ↔ q
T	T	T
T	F	F
F	T	F
F	F	T

Biconditional

- **Example**

  p: You buy an airline ticket

  q: You can take a flight

  p ↔ q: You can take a flight **iff** you buy an airline ticket

  - **True** only if you do **both** or **neither**
  - Doing only **one** or the other makes the proposition **false**

Compound Propositions

- All logical operations can be applied to build up arbitrarily complex **compound propositions**
- Any proposition can become a term inside another proposition
- **Examples**:
  - p, q, r, t are simple propositions
  - $p \vee q$ and $r \to t$ combine simple propositions
  - $(p \vee q) \to t$ and $(p \vee q) \wedge (t \vee r)$ combine simple and compound propositions into more complex compound propositions
- Parenthesis indicate the order of evaluation

Precedence of Logical Operations

- To reduce number of parentheses use precedence rules

Operation	Precedence
¬	1
∧	2
∨	3
→	4
↔	5

Examples:

- $p \vee q \wedge r$ means: $p \vee (q \wedge r)$
- $(p \vee q) \wedge r$ requires the parentheses
- $p \vee q \to \neg r$ means: $(p \vee q) \to (\neg r)$
- $p \vee (q \to \neg r)$ requires parentheses

Logical Equivalence

- A **tautology** is a proposition that is always **true**

  Example: p ∨ ¬p

p	¬p	p ∨ ¬p
T	F	T
F	T	T

- A **contradiction** is a proposition that is always **false**

  Example: p ∧ ¬p

p	¬p	p ∧ ¬p
T	F	F
F	T	F

Equivalent Propositions

- Two propositions are **logically equivalent** if they always have the same truth value
- We write this as **p ≡ q**
- **Formally**:

  p and q are logically equivalent iff p ↔ q is a tautology

Showing Equivalence

- One way to determine equivalence is to use truth tables
- **Example**: show that ¬p ∨ q is equivalent to p → q

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

Showing Non-Equivalence

- Find at least **one row** where values **differ**
- **Example**: Show that neither the *converse* nor the *inverse* of an implication are equivalent to the implication

p	q	¬p	¬q	p → q	¬p → ¬q	q → p
T	T	F	F	T		
T	F	F	T	F		
F	T	T	F	T		
F	F	T	T	T		

Showing Equivalence

- Truth tables with many variable become cumbersome
- Use **laws of logic** to transform propositions into equivalent forms
- To prove that p ≡ q, produce a series of equivalences leading from p to q:

$$p \equiv p_1$$
$$p_1 \equiv p_2$$
$$\ldots$$
$$p_n \equiv q$$

- Each step follows one of the equivalence laws

Laws of Propositional Logic

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

Laws of Propositional Logic

Identity laws	p ∨ F ≡ p	p ∧ T ≡ p
Domination laws	p ∧ F ≡ F	p ∨ T ≡ T
Double negation	¬¬p ≡ p	
Complement laws	p ∧ ¬p ≡ F ¬T ≡ F	p ∨ ¬p ≡ T ¬F ≡ T

Laws of Propositional Logic

De Morgan's laws	¬(p ∨ q) ≡ ¬p ∧ ¬q	¬(p ∧ q) ≡ ¬p ∨ ¬q
Absorption laws	p ∨ (p ∧ q) ≡ p	p ∧ (p ∨ q) ≡ p
Conditional identities	p → q ≡ ¬p ∨ q	p ↔ q ≡ (p → q) ∧ (q → p)

Equivalence Proofs

Example: Show that ¬(p ∨ (¬p ∧ q)) ≡ (¬p ∧ ¬ q)

Solution:

¬(p ∨ (¬p ∧ q))	given proposition
¬p ∧ ¬(¬p ∧ q)	De Morgan's law
¬p ∧ (¬¬p ∨ ¬q)	De Morgan's law
¬p ∧ (p ∨ ¬q)	Double negation law
(¬p ∧ p) ∨ (¬p ∧ ¬q)	Distributive law
F ∨ (¬p ∧ ¬q)	Complement law
(¬p ∧ ¬q) ∨ F	Commutative law
¬p ∧ ¬q	Identity law

Predicate Logic

- Propositional logic is not sufficient to express many concepts
- **Example 1** (due to Aristotle):
  - Given the statements:

    "All men are mortal"

    "Socrates is a man"
  - It follows that: "Socrates is mortal"
- This can't be represented in propositional logic
- Need formalism to express **objects** and **properties** of objects

Predicate Logic

- **Example 2**:
  - Statements such as "x is a perfect square" are **not** propositions
  - The truth value depends on the value of x
  - I.e., the truth value is a **function** of x
- We need a more powerful formalism: **Predicate logic**

Predicate Logic

- **Variables**: x, y, z, …
  These represent **objects**, not propositions
- Variables take on **values** from a given **domain**
  This is the set of **all possible values** a variable may take
- **Predicates**: P, Q, …
  These express **properties** of objects
- **Example**:
  - let x be an integer (domain)
  - let P denote the property "is a perfect square"
  - then P(x) means "x is a perfect square"

Predicate Logic

- Predicates are **generalizations** of propositions
- They are **functions** that return T or F depending on their variables
- They **become** propositions (have truth values) when their variables are replaced by actual values
- **Example**:
  - let x be an integer
  - let P denote the property "is a perfect square"
  - P(9) is a **true** proposition, P(8) is a **false**

Predicate Logic

- A predicate can depend on more than one variable
- **Examples**:
  - Let P(x, y) denote "$x > y$"; then:

    P(–3, –3) ≡ (–3 > –3) is F

    P(1, 0) ≡ (1 > 0) is T

  - Let R(x, y, z) denote "$x + y = z$"; then:

    R(2, –1, 5) ≡ 2 – 1 = 5 is F

    R(3, 4, 7) ≡ 3 + 4 = 7 is T

    R(1, 3, z) ≡ 1 + 3 = z is **not** a proposition

Predicate Logic

- **Logical operations** from propositional logic carry over to predicate logic
- **Example**:

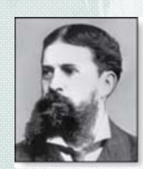
  If P(x) denotes "x > 0," then:
  - $P(3) \vee P(-1) \equiv (3 > 0) \vee (-1 > 0) \equiv T \vee F \equiv T$
  - $P(3) \wedge P(-1) \equiv F$
  - $P(3) \wedge P(y)$ is **not** a proposition

    It **becomes** a proposition when y is assigned a **value**

    or when used with **quantifiers**

Quantifiers

- **Quantifiers** express the meaning of the words **all** and **some**:
  - "All men are mortal"
  - "Some cats do not have fur"
- The two most important quantifiers are:
  - **Universal** Quantifier, **"**For all," symbol: ∀
  - **Existential** Quantifier, "There exists," symbol: ∃



Charles Peirce
(1839-1914)

Quantifiers

- Quantifiers are applied to values in a given **domain U:**
- **∀x P(x)** asserts that P(x) is T for **every** x in U

  P(x) is a **universally quantified statement**
- **∃x P(x)** asserts that P(x) is T for **some** x (at least one) in U

  P(x) is an **existentially quantified statement**
- The truth values of quantifiers depend on both the **function** P(x) and the **domain** U

Universal Quantifier

Examples:

- If P(x) denotes "x > 0" and

  U is the domain of **integers**, then

  ∀x P(x) ≡ F
- If P(x) denotes "x > 0" and

  U is the domain of **positive integers**, then

  ∀x P(x) ≡ T

Existential Quantifier

Examples:

- If P(x) denotes "$x < 0$" and
  U is **integers**, then
  $\exists x\, P(x) \equiv T$
- If P(x) denotes "$x < 0$" and
  U is **positive integers**, then
  $\exists x\, P(x) \equiv F$

Quantifiers

- A proposition with $\forall$ is equivalent to a **conjunction** of propositions without quantifiers

  $\forall x\, P(x) \equiv P(a_1) \wedge P(a_2) \wedge \ldots \wedge P(a_n)$ where $\{a_1, a_2, \ldots, a_n)$ is the domain

- A proposition with $\exists$ is equivalent to a **disjunction** of propositions without quantifiers

  $\exists x\, P(x) \equiv P(a_1) \vee P(a_2) \vee \ldots \vee P(a_n)$ where $\{a_1, a_2, \ldots, a_n)$ is the domain

- Even if the domains are **infinite**, we can still think of the quantifiers in this way

Quantified Statements

- **Logical operations** from propositional logic ($\neg \wedge \vee \rightarrow \leftrightarrow$) can also be applied to quantified statements
- **Example**:

  If P(x) denotes "x > 0" and Q(x) denotes "x is even" then:

  $\exists x\ (P(x) \wedge Q(x)) \equiv T$

  $\forall x\ (P(x) \rightarrow Q(x)) \equiv F$

Quantified Statements

- Quantifiers are applied **before** other logical operations
- **Example**:

  U: integers

  P(x): "x is even"

  Q(x): "x is divisible by 2"

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

  "for all integers x, if x is even then x is divisible by 2"

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

  "if all integers are even then x is divisible by 2" (x undefined)

Quantified Statements

- A variable is **free** if it can take on any value
- A variable is **bound** if it is within the scope of a quantifier
- **Examples**:

  $\exists x\ P(x)$

  x is bound

  the expression **is** a proposition

  $\exists x\ Q(x, y)$, $\exists x\ P(x) \land R(y)$

  x is bound but y is free

  these expressions **are not** propositions

Translating Quantified Statements

- Given a statement in natural language
  - determine the domain U
  - define predicates to capture the attributes of the statement
  - apply quantifiers and logical operations to form an expression

Translating Quantified Statements

Example 1: Translate: "Every student has taken a course in Java."

- The solution will depend on the choice of the domain
- **Solution 1**: U = {all students}
  define J(x) as: "x has taken a course in Java"
  translation: $\forall x\ J(x)$
- **Solution 2**: U = {all people}
  define S(x) as: "x is a student" and J(x) as before
  result: $\forall x\ (S(x) \rightarrow J(x))$.
- **Note:** $\forall x\ (S(x) \land J(x))$ is **not** correct. What does it mean?

Translating Quantified Statements

Example 2: Translate: "**Some** student has taken a course in Java."

- Solution again depends on the choice of domain
- **Solution 1**:
  If U = {all students} then $\exists x\ J(x)$
- **Solution 2**:
  If U = {all people} then $\exists x\ (S(x) \land J(x))$
- **Note:** $\exists x\ (S(x) \rightarrow J(x))$ is **not** correct. What does it mean?

Negating Quantified Expressions

- De Morgan's Laws for negating quantifiers:

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

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

- What is the intuition behind these laws?

Negating Quantified Expressions

- **Example 1**: "Every student has taken a course in Java"
  U = {students}
  $J(x)$: "x has taken a course in Java"
  statement: $\forall x\ J(x)$
- **Negation**: "It is not the case that every student has taken Java"
  $\neg\forall x\ J(x)$
- This implies that: "There **is** a student who has **not** taken Java"
  $\exists x\ \neg J(x)$

Negating Quantified Expressions

- **Example 2**: "There is a student who has taken a course in Java"
  $\exists x\, J(x)$
- **Negation**: "It is not true that there is a student who has taken Java"
  $\neg\exists x\, J(x)$
- This implies that: "**No** student has taken Java"
  $\forall x\, \neg J(x)$

Nested Quantifiers

- Needed to express statements with multiple variables
- **Example 1**: quantifiers of the **same** type
  "x + y = y + x for all real numbers"
  $\forall x \forall y\, (x + y = y + x)$
  where the domains of x and y are real numbers
- **Example 2**: quantifiers of **different** type
  "Every real number has an inverse"
  $\forall x \exists y\, (x + y = 0)$
  where the domains of x and y are real numbers

Nested Quantifiers

Nested Quantifiers	When True
∀x∀y P(x, y)	P(x, y) is true for every pair x, y
∃x∃y P(x, y)	There is a pair x, y for which P(x, y) is true
∀x∃y P(x, y)	For every x there is a y for which P(x, y) is true
∃x∀y P(x, y)	There is an x for which P(x, y) is true for every y

Order of Quantifiers

- With quantifiers of the **same** type, order does **not** matter
- **Example 1:** nested universal quantifiers

  U = {real numbers}

  P(x, y): "x + y = y + x"

  P(x, y) is true for **any pair** of x and y

  I.e., ∀x∀y P(x, y) ≡ T

  ≡ ∀y∀x P(x, y) ≡ T

Order of Quantifiers

- **Example 2:** nested existential quantifiers

  U = {integers}

  P(x, y, z): "$x^2 + y^2 = z^2$"

  P(x, y, z) is true for (3, 4, 5)

  I.e., $\exists x \exists y \exists z\ P(x, y, z)$

  $\equiv \exists z \exists y \exists x\ P(x, y, z)$

  $\equiv \exists x \exists z \exists y\ P(x, y, z) \equiv \ldots \equiv T$

Order of Quantifiers

- With quantifiers of **different** type, order **does** matter
- **Example**

  U = {real numbers}

  Q(x, y): "x + y = 0"

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

  because for every x there is always an inverse y

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

  because for a given y not every x will add up to zero

Negating Nested Quantifiers

- Apply De Morgan's laws **successively** from left to right
- **Example**:

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

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

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

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

- **In general:**
  - move negation to the right past all quantifiers
  - replace each ∃ with ∀, and vice versa

Moving Quantifiers

- A quantifier can be moved to the left past expressions that do not include the quantified variable
- **Example**

  $\forall x \exists y\ (\neg L(x, y) \wedge \boldsymbol{\forall z}\ (\ (z \neq y) \rightarrow \neg L(x, z)\ )\ )$

  $\forall x \exists y \boldsymbol{\forall z}\ (\neg L(x, x) \wedge (\ (z \neq y) \rightarrow \neg L(x, z)\ )\ )$

Translating Complex Statements

Follow these steps as appropriate:

1. Determine **domains** of variables
2. **Rewrite** the statement to make “for all” and “there exists” explicit
3. Introduce **variables** and define **predicates**
4. Introduce **quantifiers** and **logical operations**

Translating Complex Statements

Example 1: A mathematical statement

“The sum of two positive integers is always positive”

1. The domain:
   positive integers
2. Rewrite the statement:
   “For every two positive integers, their sum is positive”
3. Introduce variables and predicate:
   “For all positive integers x and y, x + y is positive”
4. Introduce quantifiers and logical operations:
   $\forall x \forall y\ ((x > 0) \land (y > 0) \rightarrow (x + y > 0))$

Translating Complex Statements

Example 2: "Brothers are siblings"

1. Domain: all people
2. Rewrite:
   "for any two people, if they are brothers then they are siblings"
3. Variables and predicates:
   B(x, y): "x and y are brothers"
   S(x, y): "x and y are siblings"
4. Quantifiers and logical operations:
   $\forall x \forall y\ (B(x, y) \rightarrow S(x, y))$

Translating Complex Statements

- **Example 3**: "Everybody loves somebody"
  Domain: all people
  Variables and Predicate: L(x, y): "x loves y"
  Expression: $\forall x \exists y\ L(x, y)$
- **Example 4**: "There is someone who is loved by everyone"
  $\exists y \forall x\ L(x, y)$
- **Example 5**: "There is someone who loves someone"
  $\exists x \exists y\ L(x, y)$
- **Example 6**: "Everyone loves himself/herself"
  $\forall x\ L(x, x)$

Excluding Self

Previous example: "Everybody loves somebody": ∀x∃y L(x, y)

- This allows for L(x, x)

	a	b	c
a	F	T	F
b	F	T	T
c	F	F	T

- How to express: "Everybody loves somebody **else**"
  (i.e., at least one other person)
- How to exclude self

Excluding Self

- "Everybody loves somebody **else**"
  ∀x∃y ((x ≠ y) ∧ L(x, y))

	a	b	c
a	F	T	F
b	F	T	T
c	F	T	T

- Everybody loves **at least one other** person
- but it does not **exclude** self: L(x, x)

Excluding Self

- "Everybody loves somebody **other than self**"

  $\forall x \exists y \, (L(x, y) \land \neg L(x, x))$

	a	b	c
a	F	T	F
b	F	F	T
c	T	T	F

  - Note: this allows multiple y's for any x
  - How to express: "**Exactly one**"

Excluding Multiples

- "Everybody loves **exactly one other** person (not self)"

  $\forall x \exists y \, (L(x, y) \land \forall z \, ((z \neq y) \rightarrow \neg L(x, z)))$

	a	b	c
a	F	T	F
b	F	F	T
c	F	T	F

- Move quantifier to the front:

  $\forall x \exists y \forall z \; (L(x, y) \land ((z \neq y) \rightarrow \neg L(x, z)))$

Exclude Self w/ Universal Quantifiers

Example: "Everybody loves everybody"

$\forall x \forall y\ L(x, y)$

	a	b	c
a	T	T	T
b	T	T	T
c	T	T	T

- How to express: "Everybody loves everybody **else**" (excluding self)

Exclude Self w/ Universal Quantifiers

Example: "Everybody loves everybody **else**"

$\forall x \forall y\ (\ (x \neq y) \rightarrow L(x, y)\ )$

	a	b	c
a	?	T	T
b	T	?	T
c	T	T	?

- Diagonal could be T or F
- How to **exclude self**

Exclude Self w/ Universal Quantifiers

Example: "Everybody loves everybody **else**" (and not self)

$\forall x \forall y\ (\ (\ (x \neq y) \rightarrow L(x, y)\ )\ \wedge \neg L(x, x)\ )$

	a	b	c
a	**F**	T	T
b	T	**F**	T
c	T	T	**F**

Logical Reasoning

- Logic allows us to formally prove logical statements
- An **argument** is a sequence of propositions, where:
  - All but the final proposition are **hypotheses** (or **premises**)
  - The last statement is the **conclusion**
- The argument is **valid** if the hypotheses **imply** the conclusion

Logical Reasoning

Example

- hypotheses:
  - “All men are mortal”
  - “Socrates is a man”
- conclusion:
  - “Socrates is mortal”

Argument Forms

- To avoid natural language we use **argument forms**
  - **abstract** formulation of an argument
  - uses propositional **variables**
  - it is true regardless of its **instantiation**

Argument Forms

- **Example**

Natural language argument	Argument form
If it is raining then streets are wet It is raining ∴ Streets are wet	$p \rightarrow q$ p $\therefore q$

- Any argument that matches the form is valid

Natural language argument	Argument form
If you study then you pass the course You study ∴ You pass the course	$p \rightarrow q$ p $\therefore q$

Logical Reasoning

- **Formal notation** of an argument

$$\begin{array}{l} p_1 \\ p_2 \\ \dots \\ p_n \\ \hline \therefore c \end{array}$$

- hypotheses are above the line
- conclusion is below the line
- the symbol "∴" reads: therefore

- How can we conclude c from some combination of the p_i's

Logical Reasoning

- **Truth tables** can establish validity of an argument
  - make one column for each **variable**
  - fill in all **combinations** of T/F values
  - make one column for each **hypothesis** (fill in T/F)
  - make one column for the **conjunction** of all hypotheses
  - if the conjunction implies the conclusion, the argument is **valid**

$p \rightarrow q$

$p \vee q$

$\therefore q$

p	q	p → q	p ∨ q	(p → q) ∧ (p ∨ q)
⋮	⋮	⋮	⋮	⋮

Logical Reasoning

Example

you work hard → you become successful
you are successful

∴ you worked hard

$p \rightarrow q$

q

$\therefore p$

p	q	p → q	(p → q) ∧ q
T	T	T	T
T	F	F	F
F	T	T	T
F	F	T	F

Argument is **not** valid

Rules of Inference

To avoid truth tables we construct **logical proofs**:

- sequences of steps leading from hypotheses to conclusions
- at each step we may apply **rules of inference:**
  - simple argument forms shown to be true (e.g. using truth tables)
  - they become **building blocks** in constructing incrementally more complex arguments

Modus Ponens

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

Example:

p: "It is snowing"
q: "I will study math"

"If it is snowing then I will study math"
"It is snowing"

"Therefore I will study math"

Modus Tollens

$p \to q$
$\neg q$
$\therefore \neg p$

Example:
p: “It is snowing”
q: “I will study math”

“If it is snowing, then I will study math”
“I will not study math”

“Therefore, it is not snowing”

Conjunction

p
q
$\therefore p \land q$

Example:
p: “I will study math”
q: “I will study literature”

“I will study math”
“I will study literature”

“Therefore, I will study math and literature”

Simplification

$p \land q$

$\therefore q$

Example:

p: "I will study math"
q: "I will study literature"

"I will study math and literature"

"Therefore, I will study math"

Addition

p

$\therefore p \lor q$

Example:

p: "I will study math"
q: "I will visit Las Vegas"

"I will study math"

"Therefore, I will study math or I will visit Las Vegas"

Hypothetical Syllogism

$p \rightarrow q$
$q \rightarrow r$
$\therefore p \rightarrow r$

Example:
p: "It snows"
q: "I will study math"
r: "I will get an A"

"If it snows, then I will study math"
"If I study math, then I will get an A"

"Therefore, if it snows, I will get an A"

Disjunctive Syllogism

$p \vee q$
$\neg p$
$\therefore q$

Example:
p: "I will study math"
q: "I will study literature"

"I will study math or I will study literature"
"I will not study math"

"Therefore, I will study literature"

Resolution

$$\begin{array}{r} p \lor q \\ \neg p \lor r \\ \hline \therefore q \lor r \end{array}$$

Example:

p: "It is sunny"
q: "It is raining"
r: "It is snowing"

"It is sunny or it is raining"
"It is not sunny or it is snowing"
"Therefore, it is raining or it is snowing"

Valid Arguments

- To show an argument is valid we construct a **logical proof**:
  - sequence of steps where each step can be:
    - a **hypothesis**, or
    - it must **follow** from previous steps by applying:
      - **rules of inference**, and
      - **laws of logic**
  - each step must be properly **justified**
  - if the last step is the conclusion then the argument is **valid**

Valid Arguments

- If argument is in English then first transform it into an argument **form**
  - Choose a variable for each simple proposition
  - Replace each English phrase by corresponding logical expression
  - Write argument form in formal notation

Valid Arguments

Example

Hypotheses:

- "It is not sunny and it is colder than yesterday" $\neg p \wedge q$
- "We will go swimming only if it is sunny" $r \rightarrow p$
- "If we do not go swimming, then we will take a canoe trip" $\neg r \rightarrow s$
- "If we take a canoe trip, then we will be home by sunset" $s \rightarrow t$

Conclusion: "We will be home by sunset" $\therefore t$

p: "It is sunny"
q: "It is colder than yesterday"
r: "We will go swimming"
s: "We will take a canoe trip"
t: "We will be home by sunset"

Valid Arguments

- Construct a logical proof

$\neg p \land q$
$r \rightarrow p$
$\neg r \rightarrow s$
$s \rightarrow t$
$\therefore t$

	step	justification
1.	$\neg p \land q$	Hypothesis
2.	$\neg p$	Simplification, 1
3.	$r \rightarrow p$	Hypothesis
4.	$\neg r$	Modus Tollens, 2, 3
5.	$\neg r \rightarrow s$	Hypothesis
6.	s	Modus ponens, 4, 5
7.	$s \rightarrow t$	Hypothesis
8.	t	Modus ponens, 6, 7

Rules of Inference with Quantifiers

- Before rules of inference can be applies, quantifiers must be **removed**
  - $\forall x\, P(x)$ means that P is true for **all** elements of the domain
    we can assume an **arbitrary** element c and assert that P(c) is true
  - $\exists x\, P(x)$ means that P is true for some element of the domain
    we can assume a **particular** element c and assert that P(c) is true
- Based on these observations we can derive rules of inference to **remove** or **reintroduce** quantifiers

Universal Instantiation (UI)

c is an element of the domain
$\forall x\ P(x)$
$\therefore P(c)$

- **Example**:
  domain = {all dogs}
  1. "Fido is a dog" — Hypothesis
  2. "All dogs are cuddly" — Hypothesis
  3. "Therefore, Fido is cuddly" — UI, 1, 2
- This rule **removes** the universal quantifier

Universal Generalization (UG)

c is an arbitrary element of the domain
$P(c)$
$\therefore \forall x\ P(x)$

- **Example**:
  H = {all horses}
  1. let h be an element of H — Hypothesis
  2. Herbivore(h) — Hypothesis
  3. $\forall x$ Herbivore(x) — UG, 1, 2
- This rule **introduces** a universal quantifier

Existential Instantiation (EI)

$\exists x\, P(x)$
$\therefore$ (c is a particular element of the domain – a constant) $\land$ P(c)

- **Example**:
  S = {all students in the course}
  1. "There is someone who got an A in the course" Hypothesis
  2. Introduce a constant c:
     "c is an element of S and c got an A" EI, 1
- This rule **removes** the existential quantifier

Existential Generalization (EG)

c is a particular element of the domain
P(c)
$\therefore \exists x\, P(x)$

- **Example**:
  S = {all students in the class}
  1. "Michelle is a student in the class" Hypothesis
  2. "Michelle got an A in the class" Hypothesis
  3. "Therefore, there is someone who got an A in the class" EG, 1, 2
- This rule **introduces** an existential quantifier

Returning to the Socrates Example

domain = {physical entities}
predicates:
  Man(x): "x is a man"
  Mortal(x): "x is mortal"

$\forall x\ (\text{Man}(x) \rightarrow \text{Mortal}(x))$
Man(Socrates)

∴ Mortal(Socrates)

	step	justification
1.	$\forall x\ (\text{Man}(x) \rightarrow \text{Mortal}(x))$	Hypothesis
2.	Socrates is an element of domain	Hypothesis
3.	Man(Socrates) → Mortal(Socrates)	UI, 1, 2
4.	Man(Socrates)	Hypothesis
5.	Mortal(Socrates)	MP, 3, 4