Artificial Intelligence

Propositional Logic

Knowledge Representation and Reasoning

- **Perceiving**, that is, acquiring information from environment,
- **Knowledge Representation**, that is, representing its understanding of the world,
- **Reasoning**, is the use of symbolic representations of some statements in order to derive new ones
- **Predicate logic** is the generic term for symbolic formal systems like first-order **logic**, second-order **logic**, many-sorted **logic**, or infinitary **logic**.

- **The logical language, in turn, has two aspects,**
  - Syntax
  - Semantics

Knowledge Representation and Reasoning

Syntax: Syntax specifies the symbols in the language and how they can be combined to form sentences. Hence facts about the world are represented as sentences in logic.

Semantics: The meanings of the atomic symbols of the logic, and the rules for determining the meanings of non-atomic expressions of the logic.

Valid statements or sentences in PL(Predicate Logic) are determined according to the rules of propositional syntax.

This syntax governs the combination of basic building blocks such as propositions and logical connectives.

Propositions are elementary atomic sentences.

Knowledge Representation and Reasoning

- **Propositional logic** All objects described are fixed or unique

  "John is a student" student(john)

  Here John refers to one unique person.

- **First order predicate logic**

  Objects described can be unique or variables to stand

  for a unique object

  "All students are poor"

  For All(S) [student(S) -> poor(S)]

Here S can be replaced by many different unique students.

Knowledge Representation and Reasoning

- **Predicate Logic (PL):** has three more logical notions as compared to PL.
  - ❖ Terms,
  - ❖ Predicates
  - ❖ Quantifiers
- **Term**
  - a constant (single individual or concept i.e.,5, john etc.),
  - a variable that stands for different individuals
  - n-place function f(t1, …, tn) where t1, …, tn are terms. A function is a mapping that maps n terms to a term.
- **Predicate**
  - a relation that maps n terms to a truth value true (T) or false (F).
- **Quantifiers**
  - Universal (∀) or existential(∃) quantifiers i.e. ∀ and ∃ used in conjunction with variables.

Examples

- "*x loves y*" is represented as LOVE(x, y) which maps it to true or false when x and y get instantiated to actual values.
- "*john's father loves john*" is represented as LOVE(father(john), john).
  - Here *father* is a function that maps *john* to his father.
- *x is greater than y* is represented in predicate calculus as GT(x, y).
- It is defined as follows:

GT(x, y) = , if x > y = T , otherwise F

- Symbols like **GT** and **LOVE** are called **predicates .**
  - Predicates two terms and map to T or F depending upon the values of their terms.

Examples – Cont..

- Translate the sentence *"**Every man is mortal**"* into Predicate formula.
- Representation of statement in predicate form
  - "**x is a man**" and "**MAN(x)**,
  - **x is mortal**" by **MORTAL(x)**
- Every man is mortal :

  (∀x) (MAN(x) → MORTAL(x))

Here, **∀x** is read as "**for all x**" and **→** is read as "**implies**".

Propositional Logic syntax is a combination of

- Propositions and
  - Logical connectives

Propositional

- A proposition – a sentence that can be either true or false but not both

.

- Propositions:
  - x is greater than y
  - Noam wrote this letter
  - It is raining
  - My car is painted silver
  - John and Sue have five children
  - Snow is white
  - People live on the moon

We usually denote a proposition by a letter: *p*, *q*, *r*, *s*, …

Propositional logic: Syntax

- The symbols of the language:
  - Propositional symbols (Prop): A, B, C,…
  - Connectives:
    - $\wedge$ and
    - $\vee$ or
    - $\neg$ not
    - $\rightarrow$ implies
    - $\leftrightarrow$ equivalent to
    - © xor (different than)
    - $\perp$, > False, True
  - Parenthesis:(,).
- Q1: how many different binary symbols can we define ?
- Q2: what is the minimal number of such symbols?

Logical connectives and their symbols

Logical connectives	Symbol
Not or Negation	~
And or Conjunction	&
Or or Disjunction	$\vee$
If… Then or Implication	$\rightarrow$
If and only If or Double Implication Biconditional	$\leftrightarrow$

Compound Proposition

- It is raining ***and*** wind is blowing
- The moon I made of green cheese ***or*** it is not
- **If** you study hard ***(then)*** you will be rewarded
- The sum of 10 ***and*** 20 is ***not*** 50

– T and F are special symbols having the values true and false.

PL Syntax

- The syntax of PL is recursively defined as follows
  - T and F are formulas
  - If P and Q are formulas, the following are also formula
    - (~P)
    - (P & Q)
    - (P V Q)
    - (P □Q)
    - (P □□ Q)

Examples of PL sentences

- P means "It is hot."
- Q means "It is humid."
- R means "It is raining."
- $(P \wedge Q) \rightarrow R$

  "If it is hot and humid, then it is raining"
- $Q \rightarrow P$

  "If it is humid, then it is hot"
- A better way:

  Hot = "It is hot"

  Humid = "It is humid"

  Raining = "It is raining"

Propositional logic (PL)

- A simple language useful for showing key ideas and definitions
- User defines a set of propositional symbols, like P and Q.
- User defines the **semantics** of each propositional symbol:
  - P means “It is hot”
  - Q means “It is humid”
  - R means “It is raining”
- A sentence (well formed formula) is defined as follows:
  - A symbol is a sentence
  - If S is a sentence, then $\neg$S is a sentence
  - If S is a sentence, then (S) is a sentence
  - If S and T are sentences, then ($S \vee T$), ($S \wedge T$), ($S \rightarrow T$), and ($S \leftrightarrow T$) are sentences
  - A sentence results from a finite number of applications of the above rules

Propositions: Examples

- The following are propositions
  - Today is Monday *M*
  - The grass is wet *W*
  - It is raining *R*
- The following are not propositions
  - C++ is the best language *Opinion*
  - When is the pretest? *Interrogative*
  - Do your homework *Imperative*

Truth Tables

- Truth tables are used to show/define the relationships between the truth values of
  - the individual propositions and
  - the compound propositions based on them

Propositional Semantic

- **Definition**: The value of a proposition is called its <u>truth value</u>; denoted by
  - *T* or 1 if it is true or
  - *F* or 0 if it is false
- Opinions, interrogative, and imperative are not propositions
- **Truth table**

p
0
1

Logical Connective: Negation

- $\neg p$, the negation of a proposition p, is also a proposition
- Examples:
  - Today is not Monday
  - It is not the case that today is Monday, etc.
- **Truth table**

p	$\neg p$
0	1
1	0

Logical Connective: Logical And

- The logical connective And is true only when both of the propositions are true. It is also called a **conjunction**
- Examples
  - It is raining and it is warm
  - (2+3=5) and (1<2)
  - Schroedinger’s cat is dead and Schroedinger’s is not dead.
- Truth table

p	*q*	*p^q*
0	0	
0	1	
1	0	
1	1	

Logical Connective: Logical Or

- The logical disjunction, or logical Or, is true if one or both of the propositions are true.
- Examples
  - It is raining or it is the second lecture
  - $(2+2=5) \vee (1<2)$
  - You may have
- **Truth table**

p	q	$p \wedge q$	$p \vee q$
0	0	0	
0	1	0	
1	0	0	
1	1	1	

Logical Connective: Exclusive Or

- The exclusive Or, or XOR, of two propositions is true when exactly one of the propositions is true and the other one is false
- Example
  - The circuit is either ON or OFF but not both
  - Let $ab<0$, then either $a<0$ or $b<0$ but not both
  - You may have ca
- Truth table

p	q	$p \wedge q$	$p \vee q$	$p \oplus q$
0	0	0	0	
0	1	0	1	
1	0	0	1	
1	1	1	1	

Logical Connective: Implication

- **Definition:** Let p and q be two propositions. The implication $p \rightarrow q$ is the proposition that is false when p is true and q is false and true otherwise
  - p is called the hypothesis, antecedent, premise
  - q is called the conclusion, consequence
- **Truth table**

p	q	$p \wedge q$	$p \vee q$	$p \oplus q$	$p \Rightarrow q$
0	0	0	0	0	True
0	1	0	1	1	True
1	0	0	1	1	False
1	1	1	1	0	True

Logical Connective: Implication

- The implication of $p \rightarrow q$ can be also read as
  - If p then q
  - p implies q
  - If p, q
  - p only if q
  - q if p
  - q when p
  - q whenever p
  - q follows from p
  - p is a sufficient condition for q (p is sufficient for q)
  - q is a necessary condition for p (q is necessary for p)

Logical Connective: Implication

- **Examples**
  - If you buy you air ticket in advance, it is cheaper.
  - If *x* is an integer, then $x^2 \geq 0$.
  - If it rains, the grass gets wet.
  - If 2+2=5, then all unicorns are pink.

Exercise: Which of the following implications is true?

- If -1 is a positive number, then 2+2=5

True. The premise is obviously false, thus no matter what the conclusion is, the implication holds.

- If -1 is a positive number, then 2+2=4

  True. Same as above.

- If $sin\ x = 0$, then $x = 0$

  False. x can be a multiple of π. If we let $x=2\pi$, then $sin\ x=0$ but $x\neq0$.
  The implication “if $sin\ x = 0$, then $x = k\pi$, for some k” is true.

Logical Connective: Biconditional (1)

- **Definition:** The biconditional $p \leftrightarrow q$ is the proposition that is true when p and q have the same truth values. It is false otherwise.
- Note that it is equivalent to $(p \rightarrow q) \wedge (q \rightarrow p)$
- **Truth table**

p	q	$p \wedge q$	$p \vee q$	$p \oplus q$	$p \Rightarrow q$	$p \Leftrightarrow q$
0	0	0	0	0	1	
0	1	0	1	1	1	
1	0	0	1	1	0	
1	1	1	1	0	1	

Logical Connective: Biconditional (2)

- The biconditional $p \leftrightarrow q$ can be equivalently read as
  - p if and only if q
  - p is a necessary and sufficient condition for q
  - if p then q, and conversely
  - p iff q (Note typo in textbook, page 9, line 3)
- Examples
  - $x>0$ if and only if x^2 is positive
  - The alarm goes off iff a burglar breaks in
  - You may have pudding iff you eat your meat

Exercise: Which of the following biconditionals is true?

- $x^2 + y^2 = 0$ if and only if $x=0$ and $y=0$

  True. Both implications hold

- $2 + 2 = 4$ if and only if $\sqrt{2}<2$

True. Both implications hold.

- $x^2 \geq 0$ if and only if $x \geq 0$

False. The implication "if $x \geq 0$ then $x^2 \geq 0$" holds.
However, the implication "if $x^2 \geq 0$ then $x \geq 0$" is false.
Consider $x=-1$.
The hypothesis $(-1)^2=1 \geq 0$ but the conclusion fails.

Truth tables

And

p	q	$p \cdot q$
T	T	T
T	F	F
F	T	F
F	F	F

Or

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

If . . . then

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

Not

p	$\sim p$
T	F
F	T

Truth tables II

The five logical connectives:

p	q	$p \wedge q$	$p \vee q$	$p \oplus q$	$p \Rightarrow q$	$p \Leftrightarrow q$
0	0	0	0	0	1	1
0	1	0	1	1	1	0
1	0	0	1	1	0	0
1	1	1	1	0	1	1

A complex sentence:

P	H	$P \vee H$	$(P \vee H) \wedge \neg H$	$((P \vee H) \wedge \neg H) \Rightarrow P$
False	*False*	*False*	*False*	*True*
False	*True*	*True*	*False*	*True*
True	*False*	*True*	*True*	*True*
True	*True*	*True*	*False*	*True*

Semantics

- **The semantics or meaning of a sentence is just the value true or false: that is, it is as assignment of a truth value to the sentences.**
- An interpretation for a sentence or group of sentences in an assignment of the truth value to each propositional symbol.

Semantic Rules for statements

Consider **t and f' denotes true** statements, **f and t' denotes false** statements, and **a** is any statement.

Rules Number	True Statements	False Statements
1.	T	F
2.	¯f	¯t
3.	T & ~F	f&a
4.	tVa	a&f
5.	aVt	$f \vee t'$
6.	a→t	t→f
7.	f→a	t↔f
8.	t↔ t'	f↔t
9.	f↔f'	

Example

- Let ***I*** assign true to ***P*** , false to ***Q*** and false to ***R*** in statement **((P & -Q) → R)VQ**.
- **What is the meaning of the statement?**

Answer:

- **Rule 2** gives **-Q** as **true.**
- **Rule 3** gives **(P & -Q)** as **true**.
- **Rule 6** gives **(P & -Q) →R** as **false**.
- **Rule 5** gives the statement **((P & -Q) →R)VQ** value as **false**.

Assignment

- Find the meaning of the statement

  $(^{-}P V Q) \& R \rightarrow S V (^{-}R \& Q)$

  for each of the interpretations given below.
- (a). I_1 : P is true, Q is true, R is false, S is true.
- (b). I_2 : P is true, Q is false, R is true, S is true.

Example

- Find the meaning of the following statement:
- ~(P V ~Q) & (R → S) for the interpretation given bellow:
- **I**: P is true, Q is false, R is true and S is false.

Answer:

- Rule 2 gives -**Q** as **true.**
- Rule 4 gives **(P V** -**Q)** as **true**.
- Rule 4 gives ~**(P V** -**Q)** as **false**.
- Rule 6 gives **(R → S)** as **false**.

Example

- Find the meaning of the following statement:
- ~(P V ~Q) & (R → S) for the interpretation given bellow:
- **I**: P is true, Q is false, R is true and S is false.

Answer:

- Rule 2 gives -**Q** as **true.**
- Rule 4 gives **(P V** -**Q)** as **true**.
- Rule 4 gives ~**(P V** -**Q)** as **false**.
- Rule 6 gives **(R → S)** as **false**.
- **Rule 3** gives the statement ~(P V ~Q) & (R → S) value as **false**.

Different cases of implication

- Consider the proposition $p \to q$
  - Its converse is the proposition $q \to p$
  - Its inverse is the proposition $\neg p \to \neg q$
  - Its contrapositive is the proposition $\neg q \to \neg p$

Constructing Truth Tables

- Construct the truth table for the following compound proposition

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

p	q	$p \wedge q$	$\neg q$	$((p \wedge q) \vee \neg q)$
0	0	0	1	1
0	1	0	0	0
1	0	0	1	1
1	1	1	0	1

QUICK TASK

Using truth table prove that are equivalent

$$P \leftrightarrow Q \text{ and } (P \rightarrow Q) \,\&\, (Q \rightarrow P)$$

Solution

P	Q	$P \leftrightarrow Q$
f	f	t
f	t	f
t	f	f
t	t	t

P	Q	$P \rightarrow Q$	$Q \rightarrow P$	$(P \rightarrow Q) \& (Q \rightarrow P)$
f	f	t	t	t
f	t	t	f	f
t	f	f	t	f
t	t	t	t	t

Since both have the same truth value under every interpretation,
then they **are equivalent**

Properties of Statement

- **Tautology :** A compound proposition that is always true, no matter what the truth values of the propositions that occur in it is called a **tautology**
  - P V ~P
- **Contradiction:** A compound proposition that is always false is called a **contradiction**
  - P & ~P

- **Contingency:** A proposition that is neither a tautology nor a contradiction is a **contingency**

.

- Examples
  - A simple tautology is $p \vee \neg p$
  - A simple contradiction is $p \wedge \neg p$

Properties of Statement

- **Equivalence:** Two sentences are equivalent if they have the **same truth value** under every interpretations.

- **Logical Consequences:** A sentence is a logical consequence of another if it is satisfied by all interpretations which satisfy the first.

- **A toghtology statement is satisfiable, and a contradictory statement is invalid, but the converse is not necessarily true.**

Example: On The above definitions:

- .**P V ~P** is **toghtology** since every interpretation results in a value of **true for (P V ~P).**
- **P & ~P** is a **contradiction** since every interpretation results in a value of **false for (P & ~P)**.
- **P and ~(~P)** are **equivalent** since each has the **same truth values** under every interpretation.
- **P** is a **logical consequence** of (P & Q) since any interpretation for which **(P & Q) is true**, **P is also true**.

Logical Equivalences: Example 1

- Show that (Exercise 25 from Rosen)

$$(p \to r) \vee (q \to r) \equiv (p \wedge q) \to r$$

p	q	r	$p \to r$	$q \to r$	$(p \to r) \vee (q \to r)$	$p \wedge q$	$(p \wedge q) \to r$
0	0	0					
0	0	1					
0	1	0					
0	1	1					
1	0	0					
1	0	1					
1	1	0					
1	1	1					

Inference Rules

- **The inference rules of PL provide the means to perform logical proofs or deductions.**

$$\frac{\begin{array}{c}P_1\\P_2\end{array}}{Q}$$

is a inference rule if $(P_1 \land P_2) \to Q$ is a tautology.

- Few Such Rules are as follows:
  - **Modus ponens**
  - **Chain Rule**

Modus Ponens:

From P and P → Q infer Q. This sometimes written as

$$\frac{P}{\begin{array}{c} P \rightarrow Q \\ Q \end{array}}$$

P	means: "there is a storm."
$P \rightarrow Q$	means: "if there is a storm, then the office is closed."
Q	means: "the office is closed."

Exercise: Show $[P \wedge (P \rightarrow Q)] \rightarrow Q \equiv T$.

Example For Modus Ponens:

- Given: (Joe is a father) And: (Joe is a father) → (Joe has a child)
- Conclude: (Joe has a child)

- Recall that $P \to Q \equiv \neg\, P \vee Q \equiv Q \vee \neg\, P \equiv \neg\, Q \to \neg\, P$.
- The MP rule just studied above tells us that:

$$\begin{array}{l} \neg\, Q \\ \underline{\neg\, Q \to \neg\, P} \\ \neg\, P \end{array}$$

- If we replace the $\neg\, Q \to \neg\, P$ in the above with the logically equivalent proposition $P \to Q$, then we get another implication rule:

Modus Tonens (MT) Rule:

$$\begin{array}{l} \neg\, Q \\ \underline{P \to Q} \\ \neg\, P \end{array}$$

Exercise: Show $[\neg Q \wedge (P \to Q)] \to \neg P \equiv T$.

Chain Rule hypothetical Syllogism

- Form $P \rightarrow Q$ and $Q \rightarrow R$, infer $P \rightarrow R$.

Or

$$\frac{\begin{array}{c} P \rightarrow Q \\ Q \rightarrow R \end{array}}{P \rightarrow R}$$

$P \rightarrow Q$	means	"if there is a storm, then the office is closed."
$Q \rightarrow R$	means	"if the office is closed, then I don't go to work."
$P \rightarrow R$	means	"if there is a storm, then I don't go to work."

Example for Chain Rule

- Given: (programmer likes LISP) → (programmer hates COBOL)
- **and** : (programmer hates COBOL) → (programmer likes recursion)
- **Conclude**: (programmer likes LISP) → (programmer likes recursion)

- LISP → **Lis**t **P**rocessing
- COBOL → **Co**mmon **B**usiness **O**riented Programming **L**anguage
- Prolog → **Programming in Log**ic

Logical equivalence vs. inference

By using inference rules, we can "prove" the conclusion follows from the premises. In inference, we can always replace a logic formula with another one that is logically equivalent, just as we have seen for the implication rule.

Example:

Suppose we have: $P \to (Q \to R)$ and $Q \wedge \neg R$. Use inference to show $\neg P$.

- First, we note $Q \wedge \neg R \equiv \neg(\neg Q \vee R) \equiv \neg(Q \to R)$.
- So we have the following inference:

(1)	$P \to (Q \to R)$	Premise
(2)	$Q \wedge \neg R$	Premise
(3)	$\neg(Q \to R)$	Logically equivalent to (2)
(4)	$\neg P$	Applying the second implication rule (Modus Tonens) to (1) and (3)

Conjunction and Simplification Rules

Conjunction rule

$$\frac{\begin{array}{l} P \\ Q \end{array}}{P \wedge Q}$$

Intuitively, this means when you have P and Q both being true, then $P \wedge Q$ is also true.

Simplification Rule

$$\frac{P \wedge Q}{P}$$

Intuitively, this means when you have $P \wedge Q$ being true, clearly P is also true.

3rd Disjunction rule

Resolution Rule

$$\begin{array}{l} P \lor Q \\ \underline{\neg P \lor R} \\ Q \lor R \end{array}$$

- This rule plays an important role in AI systems.
- Intuitively, it means: if P implies R and $\neg P$ implies Q (why? Where do we get these implications?), then we must have either Q or R. Clearly, this is true since one of P and $\neg P$ must be true.

Rules of Inference

Rule of Inference	Tautology	Name
$\begin{array}{l} p \\ \underline{p \rightarrow q} \\ q \end{array}$	$(p \land (p \rightarrow q)) \rightarrow q$	**Modus ponens (MP)**
$\begin{array}{l} \neg q \\ \underline{p \rightarrow q} \\ \neg p \end{array}$	$(\neg q \land (p \rightarrow q)) \rightarrow \neg p$	**Modus tonens (MT)**
$\begin{array}{l} p \rightarrow q \\ \underline{q \rightarrow r} \\ p \rightarrow r \end{array}$	$((p \rightarrow q) \land (q \rightarrow r)) \rightarrow (p \rightarrow r)$	**Hypothetical syllogism (HS)**
$\begin{array}{l} p \lor q \\ \underline{\neg p} \\ q \end{array}$	$((p \lor q) \land \neg p) \rightarrow q$	**Disjunctive syllogism (DS)**
$\begin{array}{l} \underline{p} \\ p \lor q \end{array}$	$p \rightarrow (p \lor q)$	**Addition**
$\begin{array}{l} \underline{p \land q} \\ p \end{array}$	$(p \land q) \rightarrow p$	**Simplification**
$\begin{array}{l} p \\ \underline{q} \\ p \land q \end{array}$	$((p) \land (q)) \rightarrow (p \land q)$	**Conjunction**
$\begin{array}{l} p \lor q \\ \underline{\neg p \lor r} \\ q \lor r \end{array}$	$((p \lor q) \land (\neg p \lor r)) \rightarrow (q \lor r)$	**Resolution**

Theorem

- **Theorem 4.1: The sentence s is a logical consequence of s_1 , s_2 , ... , s_n if and only if s_1 & s_2 & s_3 . . . . & $s_n \rightarrow$ s is valid/toghtology.**
- **Proof:** Theorem 4.1 can be seen by first noting that if s is a logical consequence of s_1 , s_2 , ... , s_n , then for any interpretation ***I*** in which s_1 & s_2 & s_3 , ... &s_n $\rightarrow$ s is true.
- on the other hand, if s_1 & s_2 & s_3 , ... & s_n $\rightarrow$s is totologogy, then for any interpretation ***I*** if s_1 & s_2 & s_3 , ... &s_n is true, **s is also true**.
- When **s** is a logical consequence of the set **S** = {s_1 , s_2 , ... , s_n} we will also set **S** logically implies or logically entails **S**, written

Theorem

- **Theorem 4.1: The sentence s is a logical consequence of s_1 , s_2 , ... , s_n if and only if s_1 & s_2 & s_3 . . . . & s_n → s is valid/toghtology.**

- A mathematical proof is always like:
  "If q_1 and $q_2 \ldots$ and q_k are true, then q is true."
- The propositions $q_1, \ldots, q_k$ are called the premises.
- The proposition q is called the conclusion.
- The mathematical proof is really to show that $(q_1 \wedge q_2 \ldots \wedge q_k) \rightarrow q$ is a tautology.

To do this, we can either:

1. Directly prove $(q_1 \wedge q_2 \ldots \wedge q_k) \rightarrow q - T$ by using logic equivalence rules, (which will be very long); or

2.Present a valid argument, by using logic inference rules, defined in the following slide.

Theorem

- **Theorem 4.1: The sentence s is a logical consequence of s_1 , s_2 , ... , s_n if and only if s_1 & s_2 & s_3 . . . . & s_n → s is valid/toghtology.**

Remark:

If $P_1, P_2, \ldots, P_n, Q$ is a valid argument, then we can always show:

$$[q_1 \wedge q_2 \wedge \cdots \wedge q_k] \rightarrow q - T \tag{1}$$

by using logic equivalence rules.
So a valid argument is just a shorter way to prove (1) is a tautology by using logic equivalence rules.

Theorem

- **Theorem 4.2:** The sentence s is a logical consequence of s_1 , s_2 , ... , s_n if and only if **s_1 & s_2 & s_3 , ... & s_n & ~s** is **inconsistent**.

- **Proof:** The proof of theorem 4.2 follows directly from theorem 4.1 since **s** is a logical consequence of **s_1 , s_2 , ... , s_n** if if **s_1 & s_2 & s_3 , ... &s_n → s** is **toghtology** , that is, if and only if
- ~(s_1 & s_2 & s_3 , ... &s_n → s) is inconsistent.

Conti …..

- But

$\sim(s_1 \,\&\, s_2 \,\&\, s_3 \,,\ \ldots.. \,.\, \&s_n \rightarrow s) = \sim(\sim(s_1 \,\&\, s_2 \,\&\, s_3 \,,\ \ldots.. \,.\, \&s_n) \vee s)$

[**By Conditional Elimination**]

$= \sim\sim(s_1 \,\&\, s_2 \,\&\, s_3 \,,\ \ldots.. \,.\, \&s_n) \,\&\, \sim s)$

[**By De Morgan's Law**]

$= s_1 \,\&\, s_2 \,\&\, s_3 \,,\ \ldots.. \,.\, \&s_n \,\&\, \sim s$

- When s is a logical consequence of s_1 , s_2 , …… , s_n , the formula $s_1 \,\&\, s_2 \,\&\, s_3 \,,\ \ldots.. \,.\, \&s_n \rightarrow s$ is called a theorem, with **s is the conclusion.**

Example:

Premises:

a. "It's not sunny and it's colder than yesterday"
b. "We will go swimming only if it's sunny."
c. "If we don't go swimming then we will take canoe trip."
d. "If we take a canoe trip, then we will be home by sunset."

Conclusion: "We will be home by sunset." t.

Consider the following propositions:

p: It's sunny this afternoon.
q: It's colder than yesterday.
r: We will go swimming.
s: We will take a canoe trip.
t: We will be home by sunset.

Consider the following propositions:

p: It's sunny this afternoon.

q: It's colder than yesterday.

r: We will go swimming.

s: We will take a canoe trip.

t: We will be home by sunset.

Example:

Premises:

a. "It's not sunny and it's colder than yesterday" $\neg p \wedge q$

b. "We will go swimming only if it's sunny." $r \rightarrow p$

c. "If we don't go swimming then we will take canoe trip." $\neg r \rightarrow s$

d. "If we take a canoe trip, then we will be home by sunset." $s \rightarrow t$

Conclusion: "We will be home by sunset." t.

(1)	$\neg p \wedge q$	Premise
(2)	$\neg p$	Simplification rule using (1)
(3)	$r \rightarrow p$	Premise
(4)	$\neg r$	MT using (2) (3)
(5)	$\neg r \rightarrow s$	Premise
(6)	s	MP using (4) (5)
(7)	$s \rightarrow t$	Premise
(8)	t	MP using (6) (7)

This is a valid argument showing that from the premises (a), (b) and (d), we can prove the conclusion t.

Example:

Suppose $P \to Q$; $\neg P \to R$; $Q \to S$. Prove that $\neg R \to S$.

(1)	$P \to Q$	Premise
(2)	$\neg P \vee Q$	Logically equivalent to (1)
(3)	$\neg P \to R$	Premise
(4)	$P \vee R$	Logically equivalent to (3)
(5)	$Q \vee R$	Apply resolution rule to (2)(4)
(6)	$\neg R \to Q$	Logically equivalent to (5)
(7)	$Q \to S$	Premise
(8)	$\neg R \to S$	Apply HS rule to (6)(7)

Example: Suppose:

(1) If it is Saturday today, then we play soccer or basketball.
(2) If the soccer field is occupied, we dont play soccer.
(3) It is Saturday today, and the soccer field is occupied.
Prove: “we play basketball or volleyball”.

First we formalize the problem:

P: It is Saturday today.

Q: We play soccer.

R: We play basketball.

S: The soccer field is occupied.

T: We play volleyball.

Premise: $P \to (Q \vee R),\ S \to \neg Q,\ P,\ S$

Need to prove: $R \vee T$.

(1)	$P \rightarrow (Q \vee R)$	Premise
(2)	P	Premise
(3)	$Q \vee R$	Apply MP rule to (1)(2)
(4)	$S \rightarrow \neg Q$	Premise
(5)	S	Premise
(6)	$\neg Q$	Apply MP rule to (4)(5)
(7)	R	Apply DS rule to (3)(6)
(8)	$R \vee T$	Apply Addition rule to (7)

Table 4.2 lists some of the important laws of PL (Some Equivalence Laws)

Name of Laws	Statements
Idempotency	P V P = P P & P = P
Associativity	(PVQ)VR = P V (Q V R) (P & Q) & R = P & (Q & R)
Commutativity	P V Q = Q V P P & Q = Q & P P ↔Q + Q↔P
Distributivity	P & (Q V R) = (P & Q)V (P &R) P V (Q & R) = (P V Q) & (P V R)
De Morgan's Laws	~(P V Q) = ~P & ~Q ~(P & Q) = ~P V ~Q
Conditional Elimination	P → Q = ~P V Q
Bi-conditional Elimination	P ↔Q = (p → Q) & (Q →P)

Example: DeMorgans

- Prove that $\neg(p \vee q) \Leftrightarrow (\neg p \wedge \neg q)$

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

Example: Distribution

Prove that: $p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$

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

Using Logical Equivalences: Example 1

- Logical equivalences can be used to construct additional logical equivalences
- Example: Show that $(p \wedge q) \rightarrow q$ is a tautology

- Example (Exercise 17)*: Show that $\neg(p \leftrightarrow q) \equiv (p \leftrightarrow \neg q)$

Using Logical Equivalences: Example 1

- Logical equivalences can be used to construct additional logical equivalences
- Example: Show that $(p \wedge q) \rightarrow q$ is a tautology

0. $(p \wedge q) \rightarrow q$
1. $\equiv \neg(p \wedge q) \vee q$ — Implication Law on 0
2. $\equiv (\neg p \vee \neg q) \vee q$ — De Morgan's Law (1st) on 1
3. $\equiv \neg p \vee (\neg q \vee q)$ — Associative Law on 2
4. $\equiv \neg p \vee 1$ — Negation Law on 3
5. $\equiv 1$ — Domination Law on 4

Using Logical Equivalences: Example 2

- Example (Exercise 17)*: Show that $\neg(p \leftrightarrow q) \equiv (p \leftrightarrow \neg q)$
- Sometimes it helps to start with the second proposition $(p \leftrightarrow \neg q)$

0. $(p \leftrightarrow \neg q)$
1. $\equiv (p \to \neg q) \wedge (\neg q \to p)$ — Equivalence Law on 0
2. $\equiv (\neg p \vee \neg q) \wedge (q \vee p)$ — Implication Law on 1
3. $\equiv \neg(\neg((\neg p \vee \neg q) \wedge (q \vee p)))$ — Double negation on 2
4. $\equiv \neg(\neg(\neg p \vee \neg q) \vee \neg(q \vee p))$ — De Morgan's Law…
5. $\equiv \neg((p \wedge q) \vee (\neg q \wedge \neg p))$ — De Morgan's Law
6. $\equiv \neg((p \vee \neg q) \wedge (p \vee \neg p) \wedge (q \vee \neg q) \wedge (q \vee \neg p))$ — Distribution Law
7. $\equiv \neg((p \vee \neg q) \wedge (q \vee \neg p))$ — Identity Law
8. $\equiv \neg((q \to p) \wedge (p \to q))$ — Implication Law
9. $\equiv \neg(p \leftrightarrow q)$ — Equivalence Law

*See Table 8 (p 25) but you are not allowed to use the table for the proof

Using Logical Equivalences: Example 3

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

Using Logical Equivalences: Example 3

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

0. $\neg(q \rightarrow p) \vee (p \wedge q)$
1. $\equiv \neg(\neg q \vee p) \vee (p \wedge q)$ Implication Law
2. $\equiv (q \wedge \neg p) \vee (p \wedge q)$ De Morgan's & Double negation
3. $\equiv (q \wedge \neg p) \vee (q \wedge p)$ Commutative Law
4. $\equiv q \wedge (\neg p \vee p)$ Distributive Law
5. $\equiv q \wedge 1$ Identity Law

$\equiv q$ Identity Law

First Order Periodic Logic

- PL dose not allow us to make generalized statement about classes of similar object.
- FOPL is a generalization of PL
- FOPL was developed to extend the expressiveness of PL
- The syntax for FOPL, like PL, is determined by the allowable symbols and rules of combination.
- **The semantics of FOPL are determined by interpretations assigned to predicates, rather than propositions.**

All Student in CS must take Pascal

John is a CS major

It is not possible in PL to conclude that <u>john must take Pascal</u>

- Logical connectives
- Quantifier
  - Predicates
  - Variables
  - Constants
  - Functions

Syntax Of FOPL

- In FOP statement from the natural language are translated into
- symbolic structure comprised of

- **Connectives**: ~, &, V,
- **Quantifiers**: $\forall, \exists$ $\rightarrow, \leftrightarrow$ (eg. $\forall x, \exists xy$)
- **Predicate**: P,Q,R, EQUALS, MARRID
  - Predicates are also referred to as ***atomic formula*** or ***atoms*** or ***literal***
- **Constants**: a, b, c, 3.5,-21, flight-102
- **Variables** : aircraft-type, x,y,z
- **Functions**: *f(), g(), h(), father-of()*
  - $f(t_1,t_2,\ldots,t_n)$ is a n-ary function where $\mathbf{t_i}$ are terms(constants, variables or functions)

Syntax of FOPL

- **The symbols and rules of combination permitted in FOPL are defined as follows:**
- **Connectives:** There are five connective symbols:
    - ~(not or negation)
    - & (and or conjunction)
    - V (**or** or inclusive disjunction)
    - → (implication)
    - ↔ (equivalence or if and only if).

Syntax of FOPL Quantifiers

- **Universal quantification**
  - $(\forall x)P(x)$ means that P holds for **all** values of x in the domain associated with that variable
  - E.g., $(\forall x)$ dolphin(x) $\rightarrow$ mammal(x)

- **Existential quantification**
  - $(\exists x)P(x)$ means that P holds for **some** value of x in the domain associated with that variable
  - E.g., $(\exists x)$ mammal(x) $\wedge$ lays-eggs(x)
  - Permits one to make a statement about some object without naming it

- The two quantifier symbols are $\exists$ (existential quantification) and $\forall$ (universal quantification).
- Where $(\exists$**x**$)$ means for some **x** or there is an **x** . and $(\forall$**x**$)$ means for **all x**.
- When more than one variable is being quantified by the same quantifier, such as, $(\forall$**x**$)$ $(\forall$**y**$)$ $(\forall$**z**$)$, we abbreviate with a single quantifier and drop the parentheses to get $\forall$**xyz**.

Quantifiers

- Universal quantifiers are often used with **“implies” to form “rules”:**
  $(\forall x)$ student(x) $\rightarrow$ smart(x) means “All students are smart”

- Universal quantification **is *rarely* used to make blanket statements** about every individual in the world:
- **$(\forall x)$student(x)$^\wedge$smart(x) means** “Everyone in the world is a student and is smart”

- Existential quantifiers are usually **used with “and” to specify a list** of properties about an individual:
  $(\exists x)$ student(x) $^\wedge$ smart(x) means “There is a student who is smart”

- A common mistake is to represent this English sentence as the FOL sentence:
  $(\exists x)$ student(x) $\rightarrow$ smart(x)
  - But what happens when there is a person who is *not* a student?

Quantifier Scope

- Switching the order of universal quantifiers ***does not* change the meaning:**
  - $(\forall x)(\forall y)P(x,y) \leftrightarrow (\forall y)(\forall x)\ P(x,y)$

- Similarly, **you can switch the order of existential quantifiers**:
  - $(\exists x)(\exists y)P(x,y) \leftrightarrow (\exists y)(\exists x)\ P(x,y)$

- Switching the order of universals **and existentials *does* change meaning**:
  - Everyone likes someone: $(\forall x)(\exists y)$ likes(x,y)
  - Someone is liked by everyone: $(\exists y)(\forall x)$ likes(x,y)

Connections between All and Exists

We can relate sentences involving $\forall$ and $\exists$ using De Morgan's laws:

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

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

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

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

Syntax of FOPL

- **Predicates:** Predicate symbols denote relations or functional mappings from the elements of a domain **D** to the values **true** or **false**.
- Capital letters and capitalized words such as ***P***, ***Q***, ***R***, ***EQUAL***, and ***MARRIED*** are used to represent predicates.
- Like functions, predicates may have ***n*** (***n*** ≥ ***0***) terms for arguments written as $P(t_1, t_2, t_3, \ldots \ldots, t_n)$
- Where the terms t_i , i =1 , 2, 3, ... , n are defined over some domain.
- A ***0-ary*** predicate is a proposition, that is, a constant predicate.
- **Constants**, **variables**, and **functions** are referred to as **terms**, and predicates are referred to as **atomic formulas** or **atoms** for short.

Syntax of FOPL

- **Constants:** Constants are **fixed-value** terms that belong to a given domain of discourse.
- They are denoted by **numbers**, **words**, and **small letters** near the beginning of the alphabet.
- **Examples:** a , b , c , 5.256, -67, -75.65 , flight-305, john, , Marina, etc.
- **Variables:** Variables are terms that can assume **different values** over a given domain.
- They are denoted by **words** and **small letters** near the end of the alphabet.
- **Examples:** aircraft-type, individuals, x, y, and z.

Syntax of FOPL

- **Functions**:
- **which are a subset of relations where there is only one "value" for any given "input"**
- Function symbols denote relations defined on a domain **D**. They map ***n*** elements (***n≥0***) to a single element of the domain.
- Symbols *f* , *g*, & *h*, and words such as **father-of** , or **age-of**, represent functions.
- An ***n*** place (n-ary) function is written as $f(t_1, t_2, t_3, \ldots \ldots \ldots t_n)$ where the t_i are terms (constants, variables, or functions) defined over some domain. A **0-ary** function is a **constant**.
- **Brother-of, bigger-than, outside, part-of, has-color, occurs-after, owns, visits, precedes, ..**

Syntax of FOPL

- An atomic formula is a **wffs** (**well-formed formulas**) .
- If P and Q are wffs, then ~P, P & Q, P V Q, P
  P ↔ Q, ∀x P(x), and ∃x P(x) are wffs.
- Wffs are formed only by applying the above rules a finite number of times.
- **The above rules state that all wffs are formed from atomic formulas and the proper application of quantifiers and logical connections.**

Syntax of FOPL

- **Some examples of valid wffs are**
- MAN(john)
- PILOT(father-of(bill))
- $\exists$ xyz((FATHER(x,y)&FATHER(y,z)) $\rightarrow$GRANDFATHER(x,z))
- $\forall$x NUMBER(x) $\rightarrow$($\exists$y GREATER-THAN(y,x))

Quantified inference rules

- Rules of inference start to be more useful when applied to quantified statements.
- Rules for quantified statements:
  - Universal instantiation
    - $\forall x\ P(x)\ \therefore\ P(C)$
  - Universal generalization
    - $P(C) \wedge P(B) \ldots \therefore\ \forall x\ P(x)$
  - Existential instantiation
    - $\exists x\ P(x)\ \therefore P(C)$ ← **skolem constant F**
  - Existential generalization
    - $P(C)\ \therefore\ \exists x\ P(x)$

Quantified inference rules

- Prove things that are maybe less obvious. e.g.
- “Students who pass the course either do the homework or attend lecture
- “Bob did not attend every lecture;” “Bob passed the course.”

- Translate into logic as (with domain being students in the course): $\forall x(P(x) \rightarrow H(x) \vee L(x))$, $\neg L(b)$, $P(b)$.
- Then we can conclude:
  1. $\forall x(P(x) \rightarrow H(x) \vee L(x))$ [hypothesis]
  2. $P(b) \rightarrow H(b) \vee L(b)$ [Universal instantiation]
  3. $P(b)$ [hypothesis]
  4. $H(b) \vee L(b)$ [modus ponens using (2) and (3)]
  5. $\neg L(b)$ [hypothesis]
  6. $H(b)$ [Disjunctive syllogism using (4) and (5)]
- So, Bob must have done the homework.

Quantified inference rules

e.g. “Bob failed the course, but attended every lecture;” “everyone who did the homework every week passed the course;”
“if a student passed the course, then they did some of the homework.”

We want to conclude that not every student submitted every homework assignment.
Translate into logic as (domain for *s* being students in the course and *w* being weeks of the semester):

$$\neg P(b) \land \forall w(L(b,w)),$$
$$\forall s[(\forall w H(s,w)) \rightarrow P(s)]$$
$$\forall s[P(s) \rightarrow \exists w H(s,w)].$$

Quantified inference rules

- Then we can conclude:
  1. $\neg P(b) \wedge \forall w(L(b,w))$ [hypothesis]
  2. $\neg P(b)$ [simplification using (1)]
  3. $\forall s[(\forall w H(s,w)) \rightarrow P(s)]$ [hypothesis]
  4. $(\forall w H(b,w)) \rightarrow P(b)$ [universal instantiation using (3)]
  5. $\neg\forall w H(b,w)$ [modus tollens using (4) and (2)]
  6. $\exists w \neg H(b,w)$ [quantifier negation using (5)]
  7. $\exists s \exists w \neg H(s,w)$ [existential generalization using (6)]
- So, somebody didn't hand in one of the homeworks.
- We didn't use one of the hypotheses. That's okay.

In the last line, could we have concluded that $\forall s \exists w \neg H(s,w)$ using universal generalization?

- i.e. every student missed at least one homework.
- Hopefully not: there's no evidence in the hypotheses of it (intuitively).
- The problem is that b isn't just anybody in line 1 (or therefore 2, 5, 6, or 7). It's Bob.
- It's not an "arbitrary" value, so we can't apply universal generalization.

Translating English to FOL

Every gardener likes the sun.

$\forall x \; gardener(x) \rightarrow likes(x,Sun)$

You can fool some of the people all of the time.

$\exists x \; \forall t \; person(x) \wedge time(t) \rightarrow can\text{-}fool(x,t)$

You can fool all of the people some of the time.

$\forall x \; \exists t \; (person(x) \rightarrow time(t) \wedge can\text{-}fool(x,t))$

All purple mushrooms are poisonous.

$\forall x \; (mushroom(x) \wedge purple(x)) \rightarrow poisonous(x)$

Equivalent

No purple mushroom is poisonous.

$\forall x \; (mushroom(x) \wedge purple(x)) \rightarrow \neg poisonous(x)$

Clinton is not tall.

$\neg tall(Clinton)$

Syntax of FOPL

- **Some examples of statements that are not wffs are:**
- **$\forall P\ P(x) \rightarrow Q(x)$**
- /* Universal quantification is applied to the predicate P(x). This is **invalid** in FOPL. */
- **MAN(~john)**
- /*The expression is **invalid** since the term John, a constant, is negated. */
- **father-of(Q(x))**
- /* The expression is **invalid** due to it is function with a predicate argument. */
- **MARRIED(MAN,WOMAN)**
- /* The expression **fails** since it is predicate with two predicate arguments.*/

Translating between English and Logic Notation

- **E1**: All employees earning $1400 or more per year pay taxes.

- **E2**: Some employees are sick today

- **E3**: No employee earns more than the precedent

FOPL

- E(x) => x is an employee
- P(x) => x is president
- i(x) => income of x
- GE(u,v) => u is grater than v
- S(x) => x is sick today
- T(x) => x pays tax

FOPL

- **E1**: All employees earning $1400 or more per year pay taxes.

  - E(x) => x is an employee
  - P(x) => x is president
  - i(x) => income of x
  - GE(u,v) => u is grater than v
  - S(x) => x is sick today
  - T(x) => x pays tax

- E1': $\forall x((E(x) \& GE(i(x), 1400)) \rightarrow T(x))$

FOPL

- **E2**: Some employees are sick today
- E2': $\exists y(E(y) \rightarrow S(y))$
  - E(x) => x is an employee
  - P(x) => x is president
  - i(x) => income of x
  - GE(u,v) => u is grater than v
  - S(x) => x is sick today
  - T(x) => x pays tax

FOPL

- **E3**: No employee earns more than the precedent
- E3': $\forall xy((E(x) \,\&\, P(y)) \rightarrow \sim GE(i(x), i(y)))$

- E(x) => x is an employee
- P(x) => x is president
- i(x) => income of x
- GE(u,v) => u is grater than v
- S(x) => x is sick today
- T(x) => x pays tax

Assignment

- Marcus was a man.
- Marcus was a Pompeian.
- All Pompeians were Romans.
- Caesar was a ruler.
- All Pompeians were either loyal to Caesar or hated him.
- Every one is loyal to someone.
- People only try to assassinate rulers they are not loyal to.
- Marcus tried to assassinate Caesar.

Assignment

- Marcus was a man.

  Man(Marcus)

- Marcus was a Pompeian.

  Pompeian(Marcus)

- All Pompeians were Romans.

  $\forall x($ Pompeian$(x) \rightarrow$ Roman$(x)\ )$

Assignment

- Caesar was a ruler.

  Ruler(Caesar)

- All Pompeians were either loyal to Caesar or hated him. (inclusive Or)

  $\forall x$ Pompeian(x) $\rightarrow$ Loyalto(x, Caesar) $\vee$ Hate(x, Caesar)

Assignment

- Every one is loyal to someone.

  $\forall x \exists y$: Loyalto(x, y)

- People only try to assassinate rulers they are not loyal to.

$\forall x \forall y$ Person(x) $\wedge$ Ruler(y) $\wedge$ Tryassassinate(x, y) $\rightarrow$ ~Loyalto(x, y)

Assignment

- Marcus tried to assassinate Caesar.

  Tryassassinate(Marcus, Caesar)

Conversion to Clausal Form

- Resolution is one of the method by which we can do "mechanical inference" programmatically

- Resolution requires that all statement be converted into *normalized clausal* form.

- A formula is in **conjunctive normal form** (**CNF**) or **clausal normal form** if it is a conjunction of one or more clauses, where a clause is a disjunction of literals; otherwise put, it is **an AND of ORs**.

- We define **Clause** as *the disjunction of a number of literal*

- A **ground clause** is one in which no variable occur in the expression.

- A **horn Clause** is a clause with at most one positive literal.

Conversion Of FOPL sentence to Clausal Form

- **Step 1**: Eliminate all implication (□) and equivalency (□□) connectives

  (P → Q ≡ ~P ˅ Q, P □□ Q ≡ (~P ˅ Q) & (~Q ˅ P)

- **Step 2**: Move all negations (~) so that they immediately precedes an atom

  (use P in place of ~(~P) and ∃x ~P(x) in place of ~∀x P(x)

Conversion Of FOPL sentence to Clausal Form

- **Step 3**: Rename variables, if necessary, so that all quantifiers have different variable assignments

  $\forall x\ (P(x) \rightarrow \exists x\ Q(x))$ should be renamed as $\forall x\ (P(x) \rightarrow \exists y\ Q(y))$

- **Step 4**: Skolemize all existential quantifiers
- **Step 5**: Move all universal quantifiers to the left of the expression and put the expression into **CNF** form

Conversion Of FOPL sentence to Clausal Form

- **Step 6**: Eliminate all Universal quantifier and conjunctions since they are retained implicitly. (as $\forall x\ P(x) \equiv P(x)$).

- **Step 7**: The resulting expressions are clauses and the set of such expressions is said to be is clausal form

Conversion to Clause Form

1. Eliminate $\rightarrow$.

   $$P \rightarrow Q \equiv \neg P \vee Q$$

2. Reduce the scope of each $\neg$ to a single term.

   $$\neg(P \vee Q) \equiv \neg P \wedge \neg Q$$
   $$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$$
   $$\neg \forall x: P \equiv \exists x: \neg P$$
   $$\neg \exists x: p \equiv \forall x: \neg P$$
   $$\neg\neg P \equiv P$$

3. Standardize variables so that each quantifier binds a unique variable.

   $$(\forall x\ P(x)) \vee (\exists x\ Q(x)) \equiv (\forall x\ P(x)) \vee (\exists y\ Q(y))$$

Conversion to Clause Form

4. Move all quantifiers to the left without changing their relative order.

   $(\forall x P(x)) \vee (\exists y\, Q(y)) \equiv \forall x \exists y (P(x) \vee (Q(y))$

5. Eliminate $\exists$ (Skolemization).

   $\exists x\, P(x) \equiv P(c)$ Skolem constant

   $\forall x{:}\exists y\, P(x, y) \equiv \forall x P(x, f(x))$ Skolem function

6. Drop $\forall$.

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

7. Convert the formula into a conjunction of disjuncts.

   $(P \wedge Q) \vee R \equiv (P \vee R) \wedge (Q \vee R)$

8. Create a separate clause corresponding to each conjunct.
9. Standardize apart the variables in the set of obtained clauses.

Example

- $\exists x\ \forall y\ \forall z\ P(f(x), y, z) \rightarrow (\exists u\ Q(x, u)\ \&\ \exists v\ R(y, v))$

- $\exists x\ \forall y(\sim \forall z\ P(f(x), y, z) \vee (\exists u\ Q(x, u)\ \&\ \exists v\ R(y, v)))$

- $\forall y(\underline{\ \exists z \sim P(f(c), y, z)} \vee (\exists u\ Q(x, u)\ \&\ \exists v\ R(y, v)))$

- $\forall y(\underline{\sim P(f(c), y, g(y))} \vee \underline{(Q(c, h(y))}\ \&\ \underline{R(y, i(y))})))$

  A $\vee$ B & C