Artificial Intelligence

CC-421

Basics of Propositional Logic - definitions

A **formal language** is a set of words or expressions that are obtained using an alphabet and rules. The **alphabet** for a formal language is the set of symbols from which each word is constructed.

The **set of rules**, called the syntax of the language, specifies how elements of the alphabet are combined to construct words. These words are called well-formed strings of symbols.

Propositional logic consists of a formal language and semantics that give meaning to the well-formed strings, which are called **propositions.**

The alphabet of propositional logic contains the following symbols:

1. The letters of the English alphabet; that is, A, B, C, ..., Z, and each of these letters with an index (e.g., A_4).

2. The logical values True and False.

3. These special symbols:

   $\neg$ (NOT)

   $\wedge$ (AND)

   $\vee$ (OR)

   $\Rightarrow$ (IF-THEN)

   $\Leftrightarrow$ (IF AND ONLY IF)

   ( ) (GROUPING).

   The symbol $\neg$ is called a **unary connective** and the symbols $\wedge$, $\vee$, $\Rightarrow$, and $\Leftrightarrow$ are called **binary connectives**.

The rules for creating propositions are as follows:

1. All letters, all indexed letters, and the logical values True and False are propositions. They are called **atomic propositions**.
2. If A and B are propositions, then so are $\neg A$, $A \wedge B$, $A \vee B$, $A \Rightarrow B$, $A \Leftrightarrow B$, and (A). They are called **compound propositions**.

Example Let P, Q, and R stand for these statements about the world:

P: It is raining outside.

Q. The pavement is wet.

R: The sprinkler is on.

Then applying Rule 2 once using the $\vee$ connective, we have that P$\vee$R is a proposition. Applying Rule 2 a second time using the $\wedge$ connective, we obtain that Q$\wedge$P$\vee$R is a proposition. This proposition stands for the following statement about the world:

Q$\wedge$P$\vee$R: The pavement is wet and it is raining outside or the sprinkler is on. ■

Semantics

The semantics of propositional logic gives meaning to the propositions.

The semantics consists of rules for assigning either the value T (true) or F (false) to every proposition. Such an assignment is called the truth value of the proposition. If a proposition has truth value T, we say it is true; otherwise, we say it is false.

The semantics for propositional logic consist of the following rules:

1. The logical value True is always assigned the value T and the logical value False is always assigned the value F.
2. Every other atomic proposition is assigned a value T or F. The set of all these assignments constitutes a model or possible world. All possible worlds (assignments) are permissible.

3. The truth values obtained by applying each connective to arbitrary propositions are given by the following **truth tables**:

a:

A	$\neg A$
T	F
F	T

b:

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

c:

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

d:

A	B	$A \Rightarrow B$
T	T	T
T	F	F
F	T	T
F	F	T

e:

A	B	$A \Leftrightarrow B$
T	T	T
T	F	F
F	T	F
F	F	T

4. The truth value for a compound proposition such as $P \wedge Q \vee R$ is determined recursively using the above truth tables. This is done according to the following rules:

   (a) The ( ) grouping has highest precedence in that the truth value of an entire sub-proposition enclosed by ( ) is evaluated first.

   (b) The precedence order for the connectives is $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$.

   (c) Binary connectives that are the same associate from left to right.

What happens here ?

P: It is raining.

Q: Professor Neapolitan is 5 feet tall.

Example: Suppose you read the following sign in the shoe store: Your shoes may be returned within 30 days of the purchase date if they have not been worn. Let's express this statement using propositional logic and investigate when it is true. Let the following propositions stand for these statements about the world:

P: Your shoes have been worn.
Q: It has been no more than 30 days since you purchased the shoes.
R: Your shoes may be returned.

Then the statement of the store's policy concerning your shoes is expressed logically as the following proposition: ……..

A truth table for this proposition is……..

Tautologies and Logical Implication

- A proposition is called a tautology if and only if it is true in all possible worlds.
- A proposition is called a contradiction if and only if it is false in all possible worlds.

→ Show some examples of truth table of tautologies and contradictions.

- Given two propositions A and B, if A $\Rightarrow$ B is a tautology, we say that A **logically implies** B and we write A $\Rrightarrow$ B.

The following truth table shows that $A \wedge B \Rrightarrow A$:

A	B	$A \wedge B$	$A \wedge B \Rightarrow A$
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	T

The following truth table shows that $A \wedge (A \Rightarrow B) \Rrightarrow B$:

A	B	$A \Rightarrow B$	$A \wedge (A \Rightarrow B)$	$A \wedge (A \Rightarrow B) \Rightarrow B$
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

- Given two propositions A and B, if A , B is a tautology, we say that A and B are **logically equivalent** and we write A ≡ B.

The following truth table shows that $A \Rightarrow B \equiv \neg A \vee B$:

A	B	$\neg A$	$A \Rightarrow B$	$\neg A \vee B$	$A \Rightarrow B \Leftrightarrow \neg A \vee B$
T	T	F	T	T	T
T	F	F	F	F	T
F	T	T	T	T	T
F	F	T	T	T	T

- **Theorem** A ≡ B if and only if A and B have the same truth v alue in every possible world.
- **Theorem** Suppose we have a proposition A and a sub-proposition B within A. If in A we replace B by any proposition logically equivalent to B, we will obtain a proposition logically equivalent to A.

Simplify the following expression: ¬¬ P ∧ (Q ∨¬ Q)

Some Well-Known Logical Equivalences

Logical Equivalence	Name	
$A \vee \neg A \equiv \text{True}$	Excluded middle law	EM
$A \wedge \neg A \equiv \text{False}$	Contradiction law	CL
$A \vee \text{False} \equiv A$ $A \wedge \text{True} \equiv A$	Identity laws	IL
$A \wedge \text{False} \equiv \text{False}$ $A \vee \text{True} \equiv \text{True}$	Domination laws	DL
$A \vee A \equiv A$ $A \wedge A \equiv A$	Idempotent laws	IL
$A \wedge B \equiv B \wedge A$ $A \vee B \equiv B \vee A$	Commutivity law	CL
$(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$ $(A \vee B) \vee C \equiv A \vee (B \vee C)$	Associativity law	AL
$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$ $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$	Distributivity law	DL
$\neg(A \wedge B) \equiv \neg A \vee \neg B$ $\neg(A \vee B) \equiv \neg A \wedge \neg B$	De Morgan's laws	DeML
$A \Rightarrow B \equiv \neg A \vee B$	Implication elimination	IE
$A \Leftrightarrow B \equiv A \Rightarrow B \wedge B \Rightarrow A$	If and only if elimination	IFFE
$A \Rightarrow B \equiv \neg B \Rightarrow \neg A$	Contraposition law	CL
$\neg\neg A \equiv A$	Double negation	DN

Logical Arguments

Example:

Socrates is a man.

If Socrates is a man, then Socrates is mortal.

Therefore, Socrates is mortal.

→ Let the following propositions stand for these statements about the world:

P: Socrates is man.

Q: Socrates is mortal.

Then the statement “if Socrates is a man, then Socrates is mortal" is modeled by this proposition : $P \Rightarrow Q$.

An **argument** consists of a set of propositions, called the **premises,** and a proposition called the **conclusion**. We say that the premises **entail** the conclusion if in every model in which all the premises are true, the conclusion is also true. If the premises entail the conclusion, we say the argument is **sound;** otherwise we say it is a **fallacy.**
We write arguments showing the list of premises followed by the conclusion as follows:

1. A_1
2. A_2

$\vdots$

$n.$ $\underline{A_n}$

$\quad B$

We use the symbol $\models$ to denote "entails." So if the argument is sound we write

$$A_1, A_2, \ldots, A_n \models B,$$

and if it is a fallacy we write

$$A_1, A_2, \ldots, A_n \not\models B.$$

The argument concerning Socrates is as follows:

1. P
2. $\underline{\text{P}\Rightarrow\text{Q}}$

   Q

We have the following truth table concerning this argument:

P	Q	P $\Rightarrow$ Q
T	T	T
T	F	F
F	T	T
F	F	T

Because every world in which P and P$\Rightarrow$Q are both true, Q is also true, the premises entail the conclusion and the argument is sound. So P, P$\Rightarrow$Q $\models$ Q. ∎

Example : The following is a common idiom: “Where there's smoke there is fire." It is often used when one wants to conclude that an individual must be bad because bad statements are being made about the individual. Let’s investigate doing inference with the literal statement itself. Let the following propositions stand for these statements about the world:

P: There is fire.
Q: There is smoke.

Explain what happens in the following truth table:

P	Q	$P \Rightarrow Q$	$Q \wedge (P \Rightarrow Q)$	$Q \wedge (P \Rightarrow Q) \Rightarrow P$
T	T	T	T	T
T	F	F	F	T
F	T	T	T	F
F	F	T	F	T

Derivation Systems

Example: Suppose we are trying to determine how Randi earns a living. We know that Randi either writes books or helps other people to write books. We also know that if Randi helps other people to write books, then she earns her living as an editor. Finally, we know that Randi does not write books.

We **used inference rules** to reason deductively . Let the following propositions stand for these statements about the world:

P: Randi writes books.

Q: Randi helps other people to write books.

R: Randi earns her living as an editor.

We knew $P \vee Q$ and $\neg P$. Using these two facts, we concluded Q.

Drawing this conclusion from these facts makes use of the **disjunctive syllogism rule**. Next we concluded R because we knew Q and Q ⇒R were both true. This inference makes use of the **modus ponens rule**. A set of inference rules is called a **deduction system.**

Inference Rule	Name	
$A, B \vDash A \wedge B$	Combination rule	CR
$A \wedge B \vDash A$	Simplification rule	SR
$A \vDash A \vee B$	Addition rule	AR
$A, A \Rightarrow B \vDash B$	Modus ponens	MP
$\neg B, A \Rightarrow B \vDash \neg A$	Modus tolens	MT
$A \Rightarrow B, B \Rightarrow C \vDash A \Rightarrow C$	Hypothetical syllogism	HS
$A \vee B, \neg A \vDash B$	Disjunctive syllogism	DS
$A \Rightarrow B, \neg A \Rightarrow B \vDash B$	Rule of cases	RC
$A \Leftrightarrow B \vDash A \Rightarrow B$	Equivalence elimination	EE
$A \Rightarrow B, B \Rightarrow A \vDash A \Leftrightarrow B$	Equivalence introduction	EI
$A, \neg A \vDash B$	Inconsistency rule	IR
$A \wedge B \vDash B \wedge A$	"and" Commutivity rule	ACR
$A \vee B \vDash B \vee A$	"or" Commutivity rule	OCR
If $A_1, A_2, \ldots, A_n, B \vDash C$ then $A_1, A_2, \ldots, A_n \vDash B \Rightarrow C$	Deduction theorem	DT

A deduction system is **sound** if it only derives sound arguments.

Example We use the rules to derive the soundness of the argument in the previous example. Again let the following propositions stand for these statements about the world:

P: Randi writes books.

Q: Randi helps other people to write books.

R: Randi earns her living as an editor.

	Derivation	Rule
1	$P \vee Q$	Premise
2	$\neg P$	Premise
3	$Q \Rightarrow R$	Premise
4	Q	1, 2, DS
5	R	3, 4, MP

When we write ”Premise" in the Rule column, we mean that the proposition is one of our premises. When, for example, we write “1, 2, DS" in row 4, we mean we are using the premises in rows 2 and 3 and the disjunctive syllogism rule to deduce Q.

A deduction system is **complete** if it can derive every sound argument. The set of rules in the previous table is complete. However, it would not be complete if we removed the last rule called the **Deduction theorem**. Notice that this rule differs from the others. All the other rules concern arguments in which there are premises. The Deduction theorem is needed to derive arguments in which there are no premises. An argument without premises is simply a tautology.

Example: We derive that $A \models A \vee \neg A$.We use the rules in the previous table to derive its soundness.

	Derivation	Rule	Comment
1	A	Assumption	We assume A.
2	$A \vee \neg A$	1, AR	
3	$A \Rightarrow A \vee \neg A$	1, 2, DT	We now discharge A.
4	$\neg A$	Assumption	We assume $\neg A$.
5	$\neg A \vee A$	4, AR	
6	$A \vee \neg A$	5, CR	
7	$\neg A \Rightarrow A \vee \neg A$	4, 6, DT	We now discharge $\neg A$.
8	$A \vee \neg A$	3, 7, RC	

Resolution - Normal Forms

- **Definition** A **literal** is a proposition of the form P or ¬P, where P is an atomic proposition other than True or False.
- **Definition** A **conjunctive clause** is a conjunction of literals.
- **Definition** A **disjunctive clause** is a disjunction of literals
- **Definition** A proposition is in **disjunctive normal** form if it is the disjunction of conjunctive clauses
- **Definition** A proposition is in **conjunctive normal** form if it is the conjunction of disjunctive clauses.

Example These propositions are in disjunctive normal form:

$$(\mathrm{P} \wedge \mathrm{Q}) \vee (\mathrm{R} \wedge \neg\mathrm{P})$$

$$(\mathrm{P} \wedge \mathrm{Q} \wedge \neg\mathrm{R}) \vee (\mathrm{S}) \vee (\neg\mathrm{Q} \wedge \mathrm{T})\,.$$

This proposition is not in disjunctive normal form because $\mathrm{R}\vee\mathrm{S}\wedge\mathrm{Q}$ is not a conjunctive clause:

$$(\mathrm{P} \wedge \mathrm{Q}) \vee (\mathrm{R} \vee \mathrm{S} \wedge \mathrm{Q})\,.$$

These propositions are in conjunctive normal form:

$$(\mathrm{P} \vee \mathrm{Q}) \wedge (\mathrm{R} \vee \neg\mathrm{P})$$

$$(\mathrm{P} \vee \mathrm{Q} \vee \neg\mathrm{R}) \wedge (\mathrm{S}) \wedge (\neg\mathrm{Q} \vee \mathrm{T})\,.$$

This proposition is not in conjunctive normal form because $\mathrm{R}\vee\mathrm{S}\wedge\mathrm{Q}$ is not a disjunctive clause:

$$(\mathrm{P} \vee \mathrm{Q}) \wedge (\mathrm{R} \vee \mathrm{S} \wedge \mathrm{Q})\,.$$

Algorithm 2.1 Conjunctive_Normal_Form

Input: A proposition.

Output: A logically equivalent proposition in conjunctive normal form.

Procedure $Conjuctive\_Normal\_form(\mathbf{var}\ Proposition)$;

remove all "⇔" symbols using the *if and only if elimination* law;
remove all "⇒" symbols using the *implication elimination* law;
repeat
  if there are any double negations
    remove them using the *double negation* law;
  if there are any negations of non-atomic propositions
    remove them using *DeMorgan's* laws;
until the only negations are single negations of atomic propositions;
repeat
  if there are any disjunctions in which one or more terms is a conjunction
    remove them using these laws:

$$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C) \tag{2.1}$$

$$(A \wedge B) \vee C \equiv (A \vee C) \wedge (B \vee C)\,; \tag{2.2}$$

until *Proposition* is in conjunctive normal form;

Derivations Using Resolution

Theorem: Suppose we have the argument consisting of the premises A_1, A_2, …, A_n and the conclusion B. Then A_1, A_2, ..., $A_n \models$ B if and only if $A_1 \wedge A_2 \wedge \ldots A_n \wedge \neg$ B is a contradiction.

Corollary: Suppose we have the argument consisting of the premises A_1; A_2, ... A_n and the conclusion B. Then A_1, A_2, ..., $A_n \models$ B if and only if A_1, A_2, …, A_n, $\neg$ B False.

A soundness proof that uses previous Theorem or Corollary is called a **refutation.**

In a refutation, we show that if we add the negation of B to the premises, we obtain a contradiction. Because the premises are assumed to be true in the current world, the only way the entire conjunction can be false in this world is for ¬B to be false, which means B is true.

Theorem: The following rule, called **resolution**, is sound:

$$(A \lor P), (B \lor \neg P) \models A \lor B$$

where P is a literal and A and B are clauses.

When we use the resolution rule, we say that we have **resolved** the clauses $A \lor P$ and $B \lor \neg P$, and that the resolution is on P. The clause $A \lor B$ is called the resolvent.

Example We can resolve $Q \lor P$ and $R \lor \neg P$ to obtain the resolvent $Q \lor R$.

Example We can resolve $P \lor Q \lor R$ and $\neg S \lor Q$ to obtain the resolvent $P \lor R \lor :S$.

Example If we resolve P and $\neg P$, we obtain an empty clause. Because $P, \neg P \models \text{False}$, the resolvent of P and $\neg P$ is False.

When using resolution to obtain a soundness proof, we must resolve clauses in some order. A human can choose an arbitrary order with the hope of getting to the conclusion.

However, to write a program we need a strategy that results in specific steps. One such strategy is the **set of support strategy**. In this strategy the clauses are partitioned into two sets, the **auxiliary set** and the **set of support**.

The auxiliary set is formed in such a way that no two clauses in that set resolve to False. Ordinarily, the set of premises is such a set, and therefore we let the auxiliary contain all the premises, while the set of support includes clauses obtained from the negation of the conclusion. We then perform all possible resolutions where one clause is from the set of support. The set of all resolvents obtained in this way is added to the set of support. We then perform all possible resolutions where one clause is from the new set of support. This step is repeated until we derive False or until no further resolutions are possible. The set of support strategy is complete.

Resolution Algorithms

Algorithm 2.2 Set_of_Support_Resolution

Input: A set *Premises* containing the premises in an argument;
the *Conclusion* in the argument.
Output: The value True if *Premises* entail *Conclusion*; False otherwise.

Function *Premises_Entail_Conclusion* (*Premises*, *Conclusion*);
$Set\_of\_Support$ = clauses derived from the negation of *Conclusion*;
$Auxiliary\_Set$ = clauses derived from *Premises*;
$New = \{\ \}$;
repeat
    $Set\_of\_Support = Set\_of\_Support \cup New$;
    for each clause C in $Set\_of\_Support$
        for each clause D in $Auxiliary\_Set \cup Set\_of\_Support$
            $Resolvents$ = set of clauses obtained by resolving C and D;
            if False $\in Resolvents$
                return True;
            else
                $New = New \cup Resolvents$;
            endif
        endfor
    endfor
until $New \subseteq Set\_of\_Support$;
return False;