Logic

A logic has:

1. An **alphabet** that contains all the symbols of the language of the logic.
2. A **syntax** giving the rules that define the well formed expressions of the language of the logic (often called well formed formulae (wff)).
3. One or more **Rules of inference** that take one or more wff(s) and **derives** another wff.
4. A **semantics** by which meaning or an interpretation is given to the alphabet and expressions of the language.

Language of Propositional Logic

The alphabet of propositional logic (PL) has:

1. $\mathcal{P}$: countable set of propositional variables. These are usually symbolized as: $P, Q, R, P_1, P_2, \ldots, Q_1, Q_2 \ldots$ etc.
2. $\mathcal{L}$: be the set of logical symbols $\neg$, $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$.
3. Comma, parentheses of different types.

The rules for formation ($\mathcal{R}$) of a well formed formula (wff) are:

1. Every $P \in \mathcal{P}$ is a wff.
2. If $W_1,\ W_2$ are wffs then $\neg W_1$, $W_1 \vee W_2$, $W_1 \wedge W_2$, $W_1 \rightarrow W_2$, $W_1 \leftrightarrow W_2$ are wffs. Informally they stand for not, or, and, implies (if-then), bi-implies or equivalence (iff).

The language of propositional logic is the set of all wffs given the alphabet and $\mathcal{R}$.

What do we do in and with a logic?

- The major results in logic connect syntax and semantics via the rules of inference. We will discuss these later.
- In AI logic is the most widely used system for representing declarative knowledge. The inference rules allow new knowledge to be derived from already existing declarative knowledge.

Interpretation

An interpretation in PL called a boolean valuation or boolean interpretation is an assignment of truth values `true` or `false` to each propositional variable. Formally, an interpretation is a function $\mathcal{I} : \mathcal{P} \to \{\texttt{true}, \texttt{false}\}$
The truth values for wffs can be calculated using the following tables:

P	Q	$\neg P$	$P \vee Q$	$P \wedge Q$	$P \to Q$	$P \leftrightarrow Q$
`true`	`true`	`false`	`true`	`true`	`true`	`true`
`false`	`true`	`true`	`true`	`false`	`true`	`false`
`true`	`false`		`true`	`false`	`false`	`false`
`false`	`false`		`false`	`false`	`true`	`true`

For PL whenever we say interpretation we mean boolean interpretation.

Satisfiability, validity

Definition 1 (Satisfiable)

A wff W is **satisfiable** if there is some interpetation I in which W is `true`. A set of wffs S is satisfiable if there is an interpretation I such that every wff $W \in S$ is `true`.
A wff or set of wffs is **unsatisfiable** if it is not satisfiable. $\triangleleft$

Example 2

For example: the wff $P \vee Q$ is true in any interpretation where $P =$ `true` while the wff $(P \wedge \neg P)$ is unsatisfiable.

Definition 3 (Valid)

A wff is **valid** or is a **tautology** if it is `true` in all interpretations. $\triangleleft$

Example 4

For example: $(P \vee \neg P)$ is a valid wff.

Some valid wffs or tautologies

$P \vee \neg P$	Excluded middle
$(P \vee Q) \leftrightarrow \neg(\neg P \wedge \neg Q)$	De Morgan I
$(P \wedge Q) \leftrightarrow \neg(\neg P \vee \neg Q)$	De Morgan II
$\neg(\neg P) \leftrightarrow P$	Double negation
$(P \rightarrow Q) \leftrightarrow (\neg Q \rightarrow \neg P)$	Contrapositive
$(Q \rightarrow R) \rightarrow ((P \rightarrow Q) \rightarrow (P \rightarrow R))$	First syllogism
$(P \rightarrow Q) \rightarrow ((Q \rightarrow R) \rightarrow (P \rightarrow R))$	Second syllogism
$(P \rightarrow (Q \rightarrow R)) \leftrightarrow ((P \wedge Q) \rightarrow R)$	Transportation

Logical Consequence

Definition 5 (Logical consequence)

Let S be a set of wffs and W a wff. W is a logical consequence of S iff for every interpretation I if $I(S) = \texttt{true}$ then $I(W) = \texttt{true}$ $\triangleleft$

This is normally written as $S \models W$. If W is valid then we write $\models W$.

1-Derivation

Definition 6 (One step derivation, 1-derivable)

Let S be a set of wffs and $\mathcal{R}$ a set of inference rules. Then the wff W is 1-derivable from S if W is obtained from S by the application of some inference rule from $\mathcal{R}$. $\triangleleft$

Example 7

For example if $S = \{P \rightarrow Q, P\}$ and $\mathcal{R}$ contains the Modus ponens rule: $P \rightarrow Q$, $P \Rightarrow Q$ then we can 1-derive the wff Q.

Normally, S contains the theory of the domain. That is wffs that hold or are considered `true` for the domain being modelled and the wff theorem W that is 1-derived from it as another wff(theorem) that holds for the domain.
All elements of S and all tautologies are trivially 1-derived from S.

Derivation or proof

Definition 8 (Derivation or proof)

A derivation or proof of wff W from S a set of wffs and rules of inference $\mathcal{R}$ is defined as follows: Let $W_1, \ldots, W_n$ be a sequence of wffs such that for W_i, $1 \leq i \leq n$, the following holds: i) $W_i \in S$ or ii) W_i is a valid wff i.e. tautology or iii) W_i is obtained from $W_1, \ldots W_{i-1}$ by applying some rule of inference from $\mathcal{R}$ then W is the same as W_n.
This is written as: $S \vdash W$ $\triangleleft$

Derivation example

Example 9

For example: let $S = \{P\}$ and $\mathcal{R} =$ Modus ponens we wish to derive/prove $\neg Q \rightarrow (R \rightarrow P)$. The derivation/proof is:

1	P	$P \in S$
2	$P \rightarrow (R \rightarrow P)$	Tautology
3	$(R \rightarrow P)$	Modus ponens 1, 2
4	$(R \rightarrow P) \rightarrow (\neg Q \rightarrow (R \rightarrow P))$	Tautology - same schema as 2
5	$\neg Q \rightarrow (R \rightarrow P)$	Modus ponens 3, 4

Deduction theorem

The following is a fundamental theorem:

Theorem 10 (Deduction theorem)

Let S be a set of wffs and W, V be two wffs. Then:

$$S \cup \{W\} \vdash V \leftrightarrow S \vdash W \rightarrow V$$

Soundness

Definition 11 (Soundness)

A logic or more correctly the inference rules of a logic are sound if:

$$S \vdash W \rightarrow S \models W$$

$\triangleleft$

Soundness of a logic implies that from `true` wffs we cannot derive a `false` wff.

Completeness

Informally, a logic is complete if it can derive all `true` wffs.

Definition 12 (Completeness)

Let S be a set of wffs of a logic then the logic is complete if:

$$S \models W \rightarrow S \vdash W$$

$\triangleleft$

That is if W is a logical consequence of S then W is derivable from S.

Substitution theorem

This theorem allows a formula to be substituted by an equivalent formula in another formula one or more times to yield an equivalent formula.

Theorem 13 (Substitution of equivalences)

Let $W_1 \leftrightarrow W_2$ where $W_1,\ W_2$ are wffs of PL. In any wff V_1 if W_1 occurs as a sub-formula one or more times it can be substituted one or more times by W_2 to yield the wff V_2 and $V_1 \leftrightarrow V_2$.

Adequacy of $\neg, \vee, \wedge$

While we have defined 5 logical connectives as part of PL only 3 are needed. Implication ($\rightarrow$) and bi-implication ($\leftrightarrow$) can be expressed in terms of $\neg, \vee, \wedge$.
This follows from:

$$(P \rightarrow Q) \leftrightarrow \neg P \vee Q$$

$$(P \leftrightarrow Q) \leftrightarrow ((P \rightarrow Q) \wedge (Q \rightarrow P))$$

Normal forms

Since only 3 logical symbols are adequate all wffs of PL can be put in one of the following two forms:

Definition 14 (Conjunctive Normal Form - CNF)

A wff is in CNF if it is of the form:

$$C_1 \wedge C_2 \ldots \wedge C_n$$

where each C_i has the form: $D_1 \vee \ldots \vee D_m$ and each D_j is either a propositional variable or the negation of a propositional variable. That is it is a conjunct of disjuncts. This is often called **clausal form** $\triangleleft$

Definition 15 (Disjunctive normal form - DNF)

Similar to CNF but $\wedge$ and $\vee$ are interchanged.
That is it is a disjunct of conjuncts. $\triangleleft$

Literals, Conversion to CNF

Definition 16 (Literal)

A **literal** is propositional variable or the negation (also called complement) of a propositional variable. $\triangleleft$

A wff of PL can be converted into CNF by the following steps:

1. Replace $\leftrightarrow$, $\rightarrow$ using corresponding equivalences.
2. Push $\neg$ as far inside as possible (till variable is reached) using De Morgan.
3. Distribute $\vee$ over $\wedge$ and vice versa as needed. Make use of associative and commutative properties of $\vee$ $\wedge$ where needed.
4. Use tautologies like: $\neg\neg P \leftrightarrow P$, $P \vee P \leftrightarrow P$, $P \wedge (Q \vee \neg Q) \leftrightarrow P$, $P \vee (Q \wedge \neg Q) \leftrightarrow P$ etc. together with the substitution theorem to get the CNF.

Example of conversion to CNF

Convert to clausal form:
$\neg((P \vee Q) \wedge (\neg P \vee \neg Q)) \wedge R$
Use DeMorgan twice:
$(\neg(P \vee Q) \vee \neg(\neg P \vee \neg Q)) \wedge R$
$((\neg P \wedge \neg Q) \vee (\neg\neg P \wedge \neg\neg Q)) \wedge R$
Use equivalences:
$((\neg P \wedge \neg Q) \vee (P \wedge Q)) \wedge R$
Distribute $\vee$ over $\wedge$ and the reverse:
$((\neg P \vee (P \wedge Q)) \wedge (\neg Q \vee (P \wedge Q))) \wedge R$
$(((\neg P \vee P) \wedge (\neg P \vee Q)) \wedge ((\neg Q \vee P) \wedge (\neg Q \vee Q))) \wedge R$
Use equivalences:
$((\mathit{True} \wedge (\neg P \vee Q)) \wedge ((\neg Q \vee P) \wedge \mathit{True})) \wedge R$
$(\neg P \vee Q) \wedge (\neg Q \vee P) \wedge R$

Set of clauses, empty set

A CNF formula is a conjunction of disjuncts (C_is):

$$C_1 \wedge C_2 \ldots \wedge C_n$$

where each C_i is a disjunction of literals. The C_is are called clauses and since all clauses are *anded* it is often referred to as a set of clauses. So, given a set of PL wffs S we can ultimately transform it to a set of clauses S_c.
We know that $\Phi \models \{\text{All tautologies}\}$. So intuitively we can say that the empty set of clauses Φ is `true` in all interpretations since no formula in Φ is falsified in any interpretation.

S is equivalent to S_c

Note that the transformation of S to S_c is done through a series of equivalences. So, using the substitution theorem we get:

Theorem 17

$S \leftrightarrow S_c$

So, S_c is `true` in exactly those interpretations in which S is `true`.

Resolution rule for PL clausal form

We need only one inference rule, the **resolution rule** for doing proofs in clausal form PL.

Definition 18 (Resolution inference rule)

Given two clause $C_1 = P \vee C_1'$, $C_2 = \neg P \vee C_2'$ that have complementary literals, P and $\neg P$ the **resolution rule** allows us to derive the **resolvent** clause $C_r = C_1' \vee C_2'$. $\triangleleft$

Example 19

Let $C_1 = \neg P \vee Q$, $C_2 = P$, then the resolution rule gives us the relovent clause Q.
We see this is exactly the Modus Ponens inference rule, namely:
$\{P \rightarrow Q,\ P\} \vdash Q$

Soundness of PL resolution

Theorem 20 (Soundness of resolution)

The resolution rule is a sound rule of inference.
That is: If $S_c = \{C_1,\ C_2\}$ *where* $C_1 = P \vee C_1'$, $C_2 = \neg P \vee C_2'$ *and* $C_r = C_1' \vee C_2'$ *then* $S_c \underset{res}{\vdash} C_r \rightarrow S_c \models C_r$.

Proof.

We have to argue that $C_r = C_1' \vee C_2'$ is a logical consequence of S_c. Now consider an interpretation in which S_c and therefore C_1 and C_2 are `true`. This implies that if P is `true` then C_2' must be true since $\neg P$ is false and C_2 is `true`. Therefore C_r is `true`. Conversely, if P is `false` then C_1' is `true` and consequently C_r must be `true`. So in all interpretations in which S_c is `true` C_r is `true` proving $S_c \models C_r$. $\square$

Proof by resolution

Let S_c be the theory in clausal form and let T be the theorem to be proved. We construct the set $S_T = S_c \cup (\neg T)_c$ where we have taken the negation of the theorem, converted it to clausal form and added it to S_c. Now if T is indeed a logical consequence of S_c then S_T is a contradictory set of clauses. Since all elements of S_T are clauses the only way we can get a contradiction is to derive a clause C and also $\neg C$ using resolution[1]. This actually boils down to deriving P and $\neg P$ for some propositional variable P occurring in S_T since clauses are always a disjunction of literals. When we resolve P and $\neg P$ we get the empty or null clause as a resolvent. The null clause, symbolized by $\square$, represents contradiction. Since we get a contradiction when we add $\neg T$ to S_c (which is assumed to be consistent) we conclude that T is indeed a logical consequence of S_c - using *reductio ad absurdum* and soundness.

[1] This assumes completeness of resolution for PL - see later

Example

We have the following situation:

1. If Ram takes the bus and the bus is late then Ram misses his appointment.
2. Ram should not go home if Ram misses his appointment and Ram feels depressed.
3. If Ram does not get the job then Ram feels depressed and Ram should go home.

Prove: If Ram should go home and Ram takes the bus then Ram does not feel depressed if the bus is late.

Example - contd (1)

Let:

P1 = Ram takes the bus

P2 = Bus is late

P3 = Ram misses his appointment

P4 = Ram should go home

P5 = Ram feels depressed

P6 = Ram gets the job

The example can be written as:

1. $(P1 \wedge P2) \rightarrow P3$
2. $(P3 \wedge P5) \rightarrow \neg P4$
3. $\neg P6 \rightarrow (P5 \wedge P4)$
4. [Prove:] $(P4 \wedge P1) \rightarrow (P2 \rightarrow \neg P5)$

Proof by resolution

Translation of wffs to clausal form:

1.	$\neg P1 \vee \neg P2 \vee P3$	(from wff 1)
2.	$\neg P3 \vee \neg P5 \vee \neg P4$	(from wff 2)
3.	$P6 \vee P5$	(from wff 3)
4.	$P6 \vee P4$	(from wff 3)
5.	$P4$	(from neg. of theorem)
6.	$P1$	(from neg. of theorem)
7.	$P2$	(from neg. of theorem)
8.	$P5$	(from neg. of theorem)

Proof by resolution:

9.	$\neg P2 \vee P3$	(1,6)
10.	$P3$	(9,7)
11.	$\neg P3 \vee \neg P5$	(2,5)
12.	$\neg P3$	(11,8)
13.	$\square$	(10,12)

Set R^*

Define:

$$R_0(S_c) = S_c$$

$$R_1(S_c) = R_0(S_c) \cup \{C_r | C_r = resolvent(C_i, C_j),\ C_i, C_j \in R_0(S_c)\}$$

$$R_i(S_c) = R_{i-1}(S_c) \cup \{C_r | C_r = resolvent(C_j,\ C_k), C_j, C_k \in R_{i-1}(S_c)\}$$

$$\ldots$$

$$R^*(S_c) = \bigcup_{i=0}^{\infty} R_i(S_c)$$

$R^*(S_c)$ contains all clauses that can be derived by resolution from S_c.

Theorem 21

$R^*(S_c)$ is finite iff S_c is finite.

Theorem 22

If interpretation I satisifies resolvable clauses C_1, C_2 then I satisfies C_r, where $C_r = resolvent(C_1, C_2)$.

R^* and soundness

Theorem 23 (Soundness)

If $\square \in R^*(S_c)$ then S_c is unsatisfiable.

Proof.

Since $\square$ is unsatisfiable and is derived through a resolution proof starting from S_c theorem 22 implies that S_c is unsatisfiable. $\square$

Completeness theorem, a lemma I

Completeness requires that for every unsatisfiable set S_c of clauses we should be able to derive $\square$. The proof of completeness is done in two steps. The main step is the following lemma.

Lemma 24

Let S_c be an unsatisfiable set of clauses containing only literals $P_1, P_2, \ldots, P_n$. Let S_c^{n-1} be all the finite clauses derivable by resolution in which the only literals are $P_1, P_2, \ldots, P_{n-1}$. Then S_c^{n-1} is unsatisfiable.

Proof.

First note that if $\square \in S_c$ or $\square \in S_c^{n-1}$ then S_c or S_c^{n-1} is trivially unsatisfiable.

Assume S_c^{n-1} is satisfiable. This implies there exists an interpretation I such that S_c^{n-1} is satisfiable (i.e. every clause in S_c^{n-1} is true).

Let I_1, I_2 be extensions of I such that $I_1(P_n) = \texttt{true}$ and $I_2(P_n) = \texttt{false}$.

Completeness theorem, a lemma II

Since S_c is unsatisfiable $\exists C_1 \in S_c$ that is not satisfiable by I_1. This means $\neg P_n \in C_1$ since if $P_n \in C_1$ then C_1 is satisfiable by I_1 and if $P_n \notin C_1$ then $C_1 \in S_c^{n-1}$ and is satisfiable by I and therefore by I_1.
By a similar argument $\exists C_2 \in S_c$ that is not satisfied by I_2 and $P_n \in C_2$. So, C_1, C_2 have complementary literal P_n.
Now consider $C_r = \textit{resolvent}(C_1, C_2)$, clearly $C_r \in S_c^{n-1}$(by definition of S_c^{n-1}) so I satisfies C_r that is I satisfies $C_1 - \{\neg P_n\}$ which implies I_1 satisfies C_1 - a contradiction.
Similarly, I satisfies $C_2 - \{P_n\}$ which implies I_2 satisfies C_2 also a contradiction.
Therefore the assumption that S_c^{n-1} is satisfiable is incorrect and it is unsatifiable.

$\square$

Completeness theorem

Theorem 25 (Completeness)

If S_c is an unsatisfiable set of clauses then $\square$ is derivable from S_c by resolution.
Alternately, if S_c is an unsatisfiable set of clauses then $\square \in R^*(S_c)$.

Proof.

We apply lemma 24 iteratively. Let S_c contain the literals $P_1, \ldots, P_n$ then applying the lemma once we get S_c^{n-1} is unsatisfiable. Now apply the lemma to S_c^{n-1} to get S_c^{n-2}. Repeatedly applying the lemma (n times) we get S_c^0 which has no literals and is unsatisfiable that is $\square \in S_c^0$. Clearly, $S_c^0 \subset R^*(S_c)$ and therefore $\square \in R^*(S_c)$. $\square$