一、General definition of propositional logic

Definition 1. *a propositional calculus is a formal system $\mathcal{L} = \mathcal{L}(A, \Omega, Z, I)$ where:*

- *A is an alphabet, and the elements of A are called* ***proposition symbols*** *or* ***propositional variables****.*
- *Ω is a finite set of elements called operator symbols or logical connectives. Ω can be partitioned into disjoint subsets with respect to the arities of the operator.*
- *Z is the set of inference rules.*
- *I is the set of axioms.*
- *The language of $\mathcal{L}$, namely the set of formulas, is inductively defined by the following rules:*

  1 . *Base: any element of the alphabet A is a formula of $\mathcal{L}$.*

  2 . *if $p_1, p_2 \ldots, p_j$ are formulas and $f \in \Omega_j$, then $f(p_1, \ldots, p_j)$ is a formula.*

  3 . *Closed: Nothing else is a formula of $\mathcal{L}$.*

$\Omega = \{(), \rightarrow, \neg, \wedge, \vee, \top, \bot\}$($\top$, $\bot$ are regarded as 0-ary logical connectives)
The binding priority is (), $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\rightarrow$ is **right-associative**.
Note. In formal logic, the symbol $\Rightarrow$ is an alternative for $\rightarrow$.

Definition. (atomic formula in proposition logic)
atomic formulas in proposition logic are propositions symbols and **0-ary logical connectives $\top$, $\bot$.**

the aforementioned calling of propositional variables is from wiki：
https://en.wikipedia.org/wiki/Propositional_calculus#Generic_description_of_a_propositional_calculus

sometimes proposition symbols are called **statement/sentence symbols/ propositional atoms/atomic propositions(AP).**

well formed formulas in propositional logic are also called **propositions**.

Theorem 1.33 *For every well-formed propositional logic formula, the number of left brackets is equal to the number of right brackets.*

数学归纳法的证明可以见《logic in computer science》, page 45

Semantics

A *truth assignment* is a mapping τ : the set of atoms $\rightarrow \{T, F\}$.
A truth assignment τ can be extended to assign either T or F to every formula, as follows:

1. $(\neg A)^\tau = T$ iff $A^\tau = F$
2. $(A \wedge B)^\tau = T$ iff $A^\tau = T$ and $B^\tau = T$
3. $(A \vee B)^\tau = T$ iff $A^\tau = T$ or $B^\tau = T$
4. $(A \rightarrow B)^\tau = T$ iff $A^\tau = F$ or $B^\tau = T$
5. $(A \leftrightarrow B)^\tau = T$ iff $A^\tau = B^\tau$

a truth assignment is also called a **model or valuation or interpretation**.

Definition.(satisfiability and validity)

- τ *satisfies* A iff $A^\tau = T$; τ *satisfies* a set Φ of formulas iff τ satisfies A for all $A \in \Phi$. Φ is *satisfiable* iff some τ satisfies Φ; otherwise Φ is *unsatisfiable*. Similarly for A.
- A formula A is *valid* iff $A^\tau = T$ for all τ. A valid propositional formula is called a *tautology*.
- A and B are *logically equivalent* (written $A \Longleftrightarrow B$, or $A \equiv B$) iff $A^\tau = B^\tau$ for any τ.
- A is a logical consequence of Φ (written $\Phi \models A$) iff for any τ, if τ satisfies Φ, then τ satisfies A.

Definition 1.34 If, for all valuations in which all $\phi_1, \phi_2, \ldots, \phi_n$ evaluate to T, ψ evaluates to T as well, we say that

$$\phi_1, \phi_2, \ldots, \phi_n \vDash \psi$$

holds and call $\vDash$ the *semantic entailment* relation.

Proposition.
For a set of formulas Σ and two arbitrary formulas ϕ, ψ,
$\Sigma \vDash \phi \wedge \psi$ iff $\Sigma \vDash \phi$ and $\Sigma \vDash \psi$
$\Sigma \vDash \phi \vee \psi$ iff $\Sigma \vDash \phi$ or $\Sigma \vDash \psi$
$\Sigma \vDash \neg\phi$ iff $\Sigma \nvDash \phi$
$\Sigma \vDash \phi \rightarrow \psi$ iff when $\Sigma \vDash \phi$, $\Sigma \vDash \psi$.

Proposition.
A formula ϕ is not satisfiable iff $\phi \vDash \bot$

Proposition. (Semantically Proof by contradiction)[1]
Suppose ϕ is a **finite** set of formulas. Then
$\phi \vDash \psi$ iff $\phi \cup \{\neg\psi\}$ is unsatisfiable iff $\phi, \neg\psi \vDash \bot$ iff $\vDash \phi \rightarrow \psi$

Definition. (Logical or Semantic equivalence)
ϕ and ψ are said semantically equivalent or logically equivalent iff $\phi \vDash \psi$ and $\psi \vDash \phi$. In this case, we denote it by $\phi \equiv \psi$.

Equivalent statement. If for any truth assignment τ, $\phi^\tau = \psi^\tau$, then ϕ and ψ are said logically equivalent, denoted as $\phi \equiv \psi$.

e.g.
1. $(l_1 \wedge l_2 \wedge \ldots \wedge l_m) \vee p \equiv (l_1 \vee p) \wedge (l_2 \vee p) \wedge \ldots \wedge (l_m \vee p)$
2. $\phi \leftrightarrow \top \equiv \phi, \phi \leftrightarrow \bot \equiv \neg\phi, \phi \leftrightarrow \phi \equiv \top$

[1] http://logic.stanford.edu/classes/cs157/2011/lectures/lecture03.pdf

Natural Deduction[2]

rules of inference

	introduction	elimination
$\wedge$	$\dfrac{\phi \quad \psi}{\phi \wedge \psi}\wedge\mathrm{i}$	$\dfrac{\phi \wedge \psi}{\phi}\wedge\mathrm{e}_1 \qquad \dfrac{\phi \wedge \psi}{\psi}\wedge\mathrm{e}_2$
$\vee$	$\dfrac{\phi}{\phi \vee \psi}\vee\mathrm{i}_1 \qquad \dfrac{\psi}{\phi \vee \psi}\vee\mathrm{i}_2$	$\dfrac{\phi \vee \psi \quad \boxed{\begin{array}{c}\phi\\ \vdots\\ \chi\end{array}} \quad \boxed{\begin{array}{c}\psi\\ \vdots\\ \chi\end{array}}}{\chi}\vee\mathrm{e}$
$\rightarrow$	$\dfrac{\boxed{\begin{array}{c}\phi\\ \vdots\\ \psi\end{array}}}{\phi \rightarrow \psi}\rightarrow\mathrm{i}$	$\dfrac{\phi \quad \phi \rightarrow \psi}{\psi}\rightarrow\mathrm{e}$
$\neg$	$\dfrac{\boxed{\begin{array}{c}\phi\\ \vdots\\ \bot\end{array}}}{\neg\phi}\neg\mathrm{i}$	$\dfrac{\phi \quad \neg\phi}{\bot}\neg\mathrm{e}$
$\bot$	(no introduction rule for $\bot$)	$\dfrac{\bot}{\phi}\bot\mathrm{e}$
$\neg\neg$		$\dfrac{\neg\neg\phi}{\phi}\neg\neg\mathrm{e}$

Here the box means: with the assumption of top formula (this formula is not a premise for the whole deduction, but is a premise in the box, so we can use rules of inference within the box), we can derive the bottom formula.

---

[2] there can be other system. see https://math.stackexchange.com/questions/1319338/difference-between-logical-axioms-and-rules-of-inference

derived rules of inference

$$\frac{\phi \to \psi \quad \neg\psi}{\neg\phi}\ \mathrm{MT} \qquad\qquad \frac{\phi}{\neg\neg\phi}\ \neg\neg\mathrm{i}$$

$$\frac{\boxed{\begin{array}{c}\neg\phi\\ \vdots \\ \bot\end{array}}}{\phi}\ \mathrm{PBC} \qquad\qquad \frac{}{\phi \vee \neg\phi}\ \mathrm{LEM}$$

- MT: modus tollens
- PBC: proof by contradiction
- LEM: law of the excluded middle

Let's see how this works. Suppose we have a set of formulas[4] ϕ_1, ϕ_2, ϕ_3, ..., ϕ_n, which we will call *premises*, and another formula, ψ, which we will call a *conclusion*. By applying proof rules to the premises, we hope to get some more formulas, and by applying more proof rules to those, to eventually obtain the conclusion. This intention we denote by

$$\phi_1, \phi_2, \ldots, \phi_n \vdash \psi.$$

the symbol $\vdash$ is called derivation, ψ is called the **syntactic consequence** of $\phi_1, \ldots, \phi_n$.

e.g. trivial syntactic consequence:
$\neg\phi_1 \vee \phi_2 \vdash \phi_1 \to \phi_2$
$\neg(\phi_1 \vee \phi_2 \vee \ldots \vee \phi_n) \vdash \neg\phi_1 \wedge \neg\phi_2 \wedge \ldots \wedge \neg\phi_n$ and vice versa (De Morgan's law)
For proof see 《proofs》.

Proposition.(Syntactically Proof by contradiction)
For a set of formulas S and a formula ϕ, $S \vdash \phi$ iff $S \cup \{\neg\phi\} \vdash \bot$.

Definition.(Syntactic equivalence)
ϕ and ψ are said syntactically equivalent iff $\phi \vdash \psi$ and $\psi \vdash \phi$, denoted as $\phi \dashv\vdash \psi$.

二、Strong Soundness and Completeness

Our goal is

$$\phi_1, \ldots, \phi_n \vdash \psi \Leftrightarrow \phi_1, \ldots, \phi_n \vDash \psi$$

That is, if we can write down an argument which is just a sequence of symbols in nature with $\phi_1, \ldots, \phi_n$ as premises and ψ as the conclusion, then when $\phi_1, \ldots, \phi_n$ are T, ψ is also true, and vice versa.

First, it's usually easier to prove that $\phi_1, \ldots, \phi_n \vdash \psi \Rightarrow \phi_1, \ldots, \phi_n \vDash \psi$:
Proof by mathematical induction on the length(number of lines) of the natural deduction (proof) of $\phi_1, \ldots, \phi_n \vdash \psi \Rightarrow \phi_1, \ldots, \phi_n \vDash \psi$:

Theorem. (**Soundness**)
Let $\phi_1, \phi_2, \ldots, \phi_n$ and ψ be propositional logic formulas. Then

$$\phi_1, \phi_2, \ldots, \phi_n \vdash \psi \Rightarrow \phi_1, \phi_2, \ldots, \phi_n \vDash \psi$$

Base case: a one-line proof. If the proof has length 1 ($k = 1$), then it must be of the form

$$\begin{array}{lll} 1 & \phi & \text{premise} \end{array}$$

since all other rules involve more than one line. This is the case when $n = 1$ and ϕ_1 and ψ equal ϕ, i.e. we are dealing with the sequent $\phi \vdash \phi$. Of course, since ϕ evaluates to T so does ϕ. Thus, $\phi \vDash \phi$ holds as claimed.

Course-of-values inductive step: Let us assume that the proof of the sequent $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ has length k and that the statement we want to prove is true for all numbers less than k. Our proof has the following structure:

$$\begin{array}{lll} 1 & \phi_1 & \text{premise} \\ 2 & \phi_2 & \text{premise} \\ & \vdots & \\ n & \phi_n & \text{premise} \\ & \vdots & \\ k & \psi & \text{justification} \end{array}$$

(justification means the rules of inference used to derive ψ)

And then we exhaust all the cases:

1. Let us suppose that this last rule is $\wedge$i. Then we know that ψ is of the form $\psi_1 \wedge \psi_2$ and the justification in line k refers to two lines further up which have ψ_1, respectively ψ_2, as their conclusions. Suppose that these lines are k_1 and k_2. Since k_1 and k_2 are smaller than k, we see that there exist proofs of the sequents $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi_1$ and $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi_2$ with length *less than* k – just take the first k_1, respectively k_2, lines of our original proof. Using the induction hypothesis, we conclude that $\phi_1, \phi_2, \ldots, \phi_n \vDash \psi_1$ and $\phi_1, \phi_2, \ldots, \phi_n \vDash \psi_2$ holds. But these two relations imply that $\phi_1, \phi_2, \ldots, \phi_n \vDash \psi_1 \wedge \psi_2$ holds as well – why?

(Because of the truth value table for $\wedge$)

2. If ψ has shown using the rule $\vee_e$ and has the form of $\eta_1 \vee \eta_2$, then $\eta_1 \vee \eta_2$ is either a premise or an formula derived by some rules of inference during the proof, of which the line is < k. So we have $\phi_1, \ldots, \phi_n \vdash \eta_1 \vee \eta_2 \Rightarrow \phi_1, \ldots, \phi_n \vDash \eta_1 \vee \eta_2$. Besides, η_1, η_2 are the assumptions of two boxes, respectively, and their lines are both smaller than k. Note that by breaking one of these two boxes and add the assumption $\eta_1(\eta_2)$ as a premise, we can obtain $\phi_1, \ldots, \phi_n, \eta_1 \vdash \psi$ and $\phi_1, \ldots, \phi_n, \eta_2 \vdash \psi$ with k -1 lines of proof (one of the example is like the following)

1	$p \wedge q \rightarrow r$	premise
2	p	assumption
3	q	assumption
4	$p \wedge q$	$\wedge$i 2, 3
5	r	$\rightarrow$e 1, 4
6	$q \rightarrow r$	$\rightarrow$i 3–5
7	$p \rightarrow (q \rightarrow r)$	$\rightarrow$i 2–6

1	$p \wedge q \rightarrow r$	premise
2	p	premise
3	q	assumption
4	$p \wedge q$	$\wedge$i 2, 3
5	r	$\rightarrow$e 1, 4
6	$q \rightarrow r$	$\rightarrow$i 3–5

(after breaking the box and add p as a premise)

Therefore, we have $\phi_1, \ldots, \phi_n, \eta_1 \vDash \psi$, $\phi_1, \ldots, \phi_n, \eta_2 \vDash \psi$ and $\phi_1, \ldots, \phi_n \vDash \eta_1 \vee \eta_2$. With the help of truth table again, we obtain that $\phi_1, \ldots, \phi_n \vDash \psi$.

3. other cases are similar (of these two kinds).

Completeness

Theorem: If $\Gamma \models D$, then $\Gamma \vdash D$.

Idea of proof: Let $\Gamma = \{\phi_1, \ldots, \phi_n\}$, and $D = \psi$.

1. Show $\phi_1 \to (\phi_2 \to (\ldots(\phi_n \to \psi)\ldots))$ is valid
2. Show $\vdash \phi_1 \to (\phi_2 \to (\ldots(\phi_n \to \psi)\ldots))$
3. Show $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$

The core idea is:

note that
$\phi_1, \phi_2, \ldots, \phi_n, \phi_1 \to (\phi_2 \to (\ldots(\phi_n \to \psi)\ldots)) \vdash \psi$
If (2) holds, then
$\phi_1, \ldots, \phi_n \vdash \phi_1 \to (\phi_2 \to (\ldots(\phi_n \to \psi)\ldots))$
which leads to
$\phi_1, \ldots, \phi_n \vdash \psi$.

To prove (2) holds, we need to show (1) and another theorem:
Theorem. The formula η is valid implies $\vdash \eta$
To prove this theorem, we need to first prove a lemma:

Lemma. Let ϕ be a formula such that $p_1, \ldots, p_n$ are its only propositional atoms. Let l be any line number in ϕ's truth table. For $i = 1, 2, \ldots, n$, let

$$\hat{p}_i = \begin{cases} p_i & \text{if the entry in line } l \text{ of } p_i \text{ is true} \\ \neg p_i & \text{if the entry in line } l \text{ of } p_i \text{ is false} \end{cases}$$

Then we have

$$\hat{p}_1, \ldots, \hat{p}_n \vdash \phi \text{ if the entry for } \phi \text{ in line is true}$$
$$\hat{p}_1, \ldots, \hat{p}_n \vdash \neg\phi \text{ if the entry for } \phi \text{ in line is false}$$

Note the set of formulae is defined recursively and thus the structural induction can be used in the proof of this lemma. That is,

Basis step:
show that the result holds for all elements specified in the basis step of the recursive definition (here are propositional atoms).

1. If ϕ is a propositional atom p, we need to show that $p \vdash p$ and $\neg p \vdash \neg p$. These have one-line proofs.

Recursive step:
show that if the statement is true for each of the elements used to construct new elements in the recursive step of the definition, the result holds for these new elements (all the logical connectives need to be considered).
For $\neg$ [3]:

2. If ϕ is of the form $\neg\phi_1$ we again have two cases to consider. First, assume that ϕ evaluates to T. In this case ϕ_1 evaluates to F. Note that ϕ_1 has the same atomic propositions as ϕ. We may use the induction hypothesis on ϕ_1 to conclude that $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n \vdash \neg\phi_1$; but $\neg\phi_1$ is just ϕ, so we are done.
Second, if ϕ evaluates to F, then ϕ_1 evaluates to T and we get $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n \vdash \phi_1$ by induction. Using the rule $\neg\neg$i, we may extend the proof of $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n \vdash \phi_1$ to one for $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n \vdash \neg\neg\phi_1$; but $\neg\neg\phi_1$ is just $\neg\phi$, so again we are done.

So the Lemma is proved. By using this Lemma, we can prove the Theorem :
Now in the truth table of $\eta = \phi_1 \to (\phi_2 \to (\ldots(\phi_n \to \psi)\ldots))$, the entry for η is true for every line. So there are 2^n arguments for $\hat{\phi}_1, \ldots, \hat{\phi}_n \vdash \eta$, where $\hat{\phi}_i$ is either ϕ_i or $\neg\phi_i$. Then we can write an argument of the form (for convenience, just list the example of $n = 2$):

1.	$\phi_1 \vee \neg\phi_1$	LEM
2.	ϕ_1	assumption
3.	$\phi_2 \vee \neg\phi_2$	LEM
4.	ϕ_2	assumption
5.	η	since $\phi_1, \phi_2 \vdash \eta$
6.	$\neg\phi_2$	assumption
7.	η	since $\phi_1, \neg\phi_2 \vdash \eta$
8.	η	$\vee_e$, 3, 4–5,6–7
9.	$\neg\phi_1$	assumption
10.	$\vdots$	similar to above
11.	η	$\vee_e$, 1, 2 – 8, 9 – 10

[3] The rest can be viewed in page 51 of 《logic in Computer Science》.

By using mathematical induction, we know (2) holds for all n.
And this completes the proof of completeness.

For (1) ,
Using the truth table, it's easy to see that

$$\{\phi_1, \ldots, \phi_n\} \vDash \psi \Rightarrow \{\phi_1, \ldots, \phi_{n-1}\} \vDash (\phi_n \rightarrow \psi)$$

By applying this n times, (1) can be obtained. And thus the completeness is proved.

Clauses and CNF

Definition

- A literal is either an atom p or the negation of an atom $\neg p$; p and $\neg p$ are called complementary literals
- A clause is a disjunction of literals
- A formula is in conjunctive normal form if it is a conjunction of clauses

Example

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

Note:

1. $\neg p$ is called a negative literal (form), p is called a positive literal (form). [4]
2. an empty clause is usually denoted as $\square$ or $\bot$.
3. **a clause can be represented as a set of letters involved; a CNF can be represented as a set of clauses**

Proposition.

Every formula in propositional logic is logically equivalent to a CNF.

From the truth table

Constructing CNF of a formula ϕ consisting of atoms $p_1, p_2, \ldots, p_n$, with $\hat{p}_i$ having the same definition.

For each line in which the truth of ϕ is 0, we can construct a formula $\hat{p}_1 \wedge \hat{p}_2 \wedge \ldots \wedge \hat{p}_n$. The disjunction of all these formulae, denoted as ψ, has the same truth table of $\neg\phi$. So $\neg\psi$ has the same truth table as ϕ, and $\neg\psi$ is CNF.

From the grammar(also a procedure to transfer a sentence into CNF)

note that

$$\bigvee_{j=1}^{n} (l_{j,1} \wedge l_{j,2} \wedge \ldots \wedge l_{j,a_j}) \equiv \bigwedge_{i_1, i_2, \ldots, i_n} \bigvee_{j=1}^{n} l_{j,i_j}$$

and $\{\neg, \wedge\}$ is a complete set of operator symbols.

[4] So a positive literal is essentially a propositional symbol.

structural induction (using the completeness of $\{\neg, \wedge, \vee\}$)
base case: Any atom is a literal and thus a CNF.
inductive case: suppose $p = c_1 \wedge c_2 \wedge \ldots \wedge c_n$, q are two CNFs, then $p \wedge q$ is a CNF, and

$$\neg p \equiv \bigvee_{j=1}^{n} (\neg l_{j,1} \wedge \neg l_{j,2} \wedge \ldots \neg \wedge l_{j,a_j}) \equiv \bigwedge_{i_1,i_2,\ldots,i_n} \bigvee_{j=1}^{n} \neg l_{j,i_j}$$

is a CNF. (this is a process of **moving $\neg$ inwards**)

Horn Formula

Definition

- A Horn clause is a clause with at most one positive literal
- A Horn formula is a conjunction of Horn clauses

i.e. a Horn clause has the form

$$\overline{p}_1 \vee \overline{p}_2 \vee \ldots \vee \overline{p}_n \vee q \equiv (p_1 \wedge p_2 \wedge \ldots \wedge p_n) \rightarrow q$$

here $p_i\,(i = 1,2,\ldots,n)$ are atoms and q can be an atom or $\top$ / $\bot$.

An algorithm for checking satisfiability of a Horn formula

1. while there is a clause $\neg p_1 \vee \ldots \vee \neg p_k \vee q$ such that all p_i are marked but q is not, mark q
2. if there is a clause $\neg p_1 \vee \ldots \vee \neg p_k$ such that all p_i are marked, return false, otherwise, return true

Theorem

The algorithm is correct and has at most n cycles in its while-statement where n is the number of atoms in ϕ.

Proof:

- We prove that all marked p are true for all τ such that $\phi^\tau = T$

In the following statement, the term "proposition atom" and the term "positive letter" are interchangeably used.

1. We can prove that for any truth assignment τ, $\phi^{\tau} = \top$ implies that for any marked proposition atom p, $p^{\tau} = \top$
2. Using 1, we can prove that there exists a τ such that $\phi^{\tau} = \top$ implies any clause of the form $\vee_{i=1}^{k} \neg p_i$ has at least one unmarked p_i (Proof by contradiction).
3. We can also prove that if any clause of the form $\vee_{i=1}^{k} \neg p_i$ has at least one unmarked p_i, then we can construct a τ such that $\phi^{\tau} = \top$.
Let (1) $p^{\tau} = \top$ for all marked atoms, and let (2) $p_i^{\tau} = \bot$ for the aforementioned p_i. Then all the clauses with just negative letters or those of which the positive letter is marked or all the atoms of negative letters are marked are true. Finally, the rest clauses contain some(at least 1) unmarked atoms for negative letters together with a unmarked positive letter. Assign the atoms false under τ which are in negative form(maybe they have been already assigned in (1), but this doesn't matter). Then we can show that $\phi^{\tau} = \top$.