Aqil Azmi
Math 133 Final

Some Theorems of Propositional Logic

1. Proof

Theorem 1 (Deduction Theorem) *Let A and B denote a formula. B is deducible from the set of assumptions* $\Gamma \cup \{A\}$ *if and only if the implication* $A \to B$ *is deducible from* Γ*, where* Γ *denotes a set of propositions* $A_1, A_2, \ldots, A_n$*. In other words:*

$$\Gamma \cup \{A\} \vdash B \iff \Gamma \vdash A \to B$$

Proof. First, we need to define the basic structure of propositional logic.

Definition 1 (Proposition) A proposition is a statement that is either true or false, but not both. These are denoted by propositional variables such as "p", "q" and "r". We shall call these propositional variables *atomic sentences* to indicate that they are the most basic unit.

Definition 2 (Formula) A formula is a composition of atomic sentences and logical connectors. They also contain a truth-value which are evaluated according to the truth-values of each atomic sentence contained within them and according to the relationship of these truth-values over each logical connector.

Definition 3 (Logical connector) A logical connector allows multiple atomic sentences to be compounded to form formulae. In propositional logic, we will use "and", "not", "or", "if-then" "if and only if". We assume that the reader is familiar with the truth-tables for each connector.

Definition 4 (Deducibility) Let C be a proposition, and let S be a set of propositions. We say that C is *deducible* or *derivable* from S, denoted by $S \vdash C$, if and only if there exists a finite number or sequence of propositions, where each proposition is either an element of S, or is obtained from earlier propositions in the sequence by applying valid rules of inference, or a logical equivalence, such that the final proposition is C.
We also call this *syntactic entailment*, and we say that S (syntactically) *entails* C.

Definition 5 (Rule of inference) A rule of inference (or rule of deduction) is a method of deriving a conclusion from one or more premises, where the conclusion follows logically from the premise(s). In the most minimalist form of propositional logic, we only require two:

1. (Assumption)
$$\frac{\text{Assumption: } P}{\text{Conclusion: } Q}$$
2. (Modus ponens)
$$\frac{\text{Assumption: } P \quad \text{Assumption: } P \to Q}{\text{Conclusion: } Q}$$

Axiom 6 (Hilbert's Axioms) Hilbert's Axioms are three *tautologies*, which is a formula that is always true for all possible truth-value assignments to its atomic sentences. These axioms together with modus ponens is sufficient to derive all other theorems in propositional logic. The collection of

these four things is known as a *Hilbert proof system*.

Axiom 1: $\phi \to (\psi \to \phi)$

Axiom 2: $(\phi \to (\psi \to \chi)) \to ((\phi \to \psi) \to (\phi \to \chi))$

Axiom 3: $(\neg\phi \to \neg\psi) \to (\psi \to \phi)$

Now we are finally ready to prove the deduction theorem.
Since we only defined two rules of inference explicitly in our system, we can consider each case individually in applying it on $\Gamma \cup \{A\} \vdash B$.
With the assumption rule, this means that we can assume the antecedent as our hypothesis, $\Gamma \cup \{A\} \vdash B$. We can then further split it into two possibilities: $\Gamma \vdash B$ and $\{A\} \vdash B$. With the modus ponens rule, since it is uses only a conditional, we then have $\Gamma \cup \{A\} \vdash p \to q$ and $\Gamma \cup \{A\} \vdash p$ as our hypotheses. Therefore, in total, we have to consider three cases:
Case 1: $\Gamma \vdash B$
Applying Axiom 1, we would have $\Gamma \vdash B \to (A \to B)$. Applying modus ponens on both statements, we would derive $\Gamma \vdash A \to B$.
Case 2: $\{A\} \vdash B$
This means we have $B \in \{A\}$. This implies that $B = A$ since $\{A\}$ is a singleton set. Thus, $\Gamma \vdash A \to B$ can be rewritten as$\Gamma \vdash A \to A$. Since every statement implies itself, we know that this is tautologically true.
Case 3: $\Gamma \cup \{A\} \vdash p \to q$ and $\Gamma \cup \{A\} \vdash p$.
To figure out what we have to prove, we have to apply modus ponens to these two hypotheses which would yield $\Gamma \cup \{A\} \vdash q$. Then, applying the theorem we want to prove on this, our goal becomes $\Gamma \vdash A \to q$.
By applying the theorem to our two hypotheses, we would then yield two further hypotheses: $\Gamma \vdash A \to p \to q$ and $\Gamma \vdash A \to p$. Applying Axiom 2 onto the first of the two, we would then get $\Gamma \vdash (A \to p) \to (A \to q)$. Then, we can apply modus ponens to this hypotheses and to the second of the two previously to obtain our goal, $\Gamma \vdash A \to q$. ∎

Theorem 2 (Soundness Theorem) *Let* Γ *denote a set of propositions* $A_1, A_2, \ldots, A_n$. *If* Γ *syntactically implies C (i.e. C is deducible from* Γ*), then* Γ *semantically implies C. In other words:*

$$\Gamma \vdash C \text{ implies } \Gamma \vDash C$$

Proof. First, we need to define the notion of valuation more formally:

Definition 7 (Valuation) A valuation is an assignment of truth-values to an atomic sentence or a formula, denoted by $v(A) =$ true if it assigns true to a formula A, and $v(A) =$ false if it assigns false to a formula A.

Definition 8 (Semantic implication) Let A and B be two propositions. We say that A *semantically implies* B, denoted by $A \vDash B$, if and only if every truth assignment that satisfies A also satisfies B, i.e. all truth assignments that make A true also make B true. This is also knowns as the *tautological entailment* because it is equivalent to saying that the formula $A \to B$ is a tautology. In other words:

$$A \vDash B \iff \forall v, (v(A) = true \to v(B) = true)$$

Since we only defined two rules of inference explicitly in our system, we can consider each case individually in applying it on $\Gamma \vdash C$.

With the assumption rule, this means that we can assume the antecedent as our hypothesis, $\Gamma \vdash C$. With the modus ponens rule, since it uses only a conditional, we then have $\Gamma \vdash p \to q$ and $\Gamma \vdash p$ as our hypotheses. Therefore, we have to consider two cases:
Case 1: $\Gamma \vdash C$
This simply means that the proof of C relies on an assumption in Γ. This implies that there exists some truth assignment that satisfies every member of Γ and the assumption, which implies that this truth assignment also satisfies C. In other words, $\exists v, v(\Gamma) = true \to v(C) = true$. This would hold true for any valuation of v, thus we can generalise to have $\forall v, v(\Gamma) = true \to v(C) = true$. By the definition of semantic implication, this means we have $\Gamma \vDash C$.
Case 2: $\Gamma \vdash p \to q$ and $\Gamma \vdash p$
With the same reasoning as above, this implies that there exists some truth assignment that satisfies every member of Γ and the assumption which also satisfies $p \to q$. Applying the same reasoning, to the second hypothesis, we have some truth assignment that satisfies every member of Γ and the assumption which also satisfies p. In other words, we have $\exists v, v(\Gamma) = true \to v(p \to q) = true$. Again, with the same reasoning we would have $\exists v, v(\Gamma) = true \to v(p) = true$. Applying modus ponens to both $v(p \to q) = true$ and $v(p) = true$, we would then have $v(q) = true$ for some v. Thus, we can generalise this for all v and have $\forall v, v(\Gamma) = true \to v(q) = true$. Therefore, we would have $\Gamma \vDash q$. ∎

Theorem 3 (Completeness Theorem) *Let* Γ *denote a set of propositions* $B_1, B_2, \ldots, B_n$. *If* Γ *semantically implies A, then* Γ *syntactically implies A (i.e. A is deducible from* Γ*). In other words:*

$$\Gamma \vDash A \text{ implies } \Gamma \vdash A$$

Proof. This proof uses a proof construction method and relies heavily on Theorem 1 (Deduction Theorem). This proof is due to Kalmar, 1935.

Definition 9 Let A be a formula and $b_1, b_2, \ldots, b_n$ all be atomic sentences that occur in A. We define, for $A, b_1, b_2, \ldots, b_n$ and v, the corresponding formulae $A', B_1, B_2, \ldots, B_n$ as follows:

$$A' = \begin{cases} A & \text{if } v^*(A) = true \\ \neg A & \text{if } v^*(A) = false \end{cases}$$

and also

$$B_i = \begin{cases} b_i & \text{if } v(b_i) = true \\ \neg b_i & \text{if } v(b_i) = false \end{cases}$$

for $i = 1,2,...,n$.

Lemma 10 (Main Lemma) For any formula A and an assignment variable v, if $A', B_1, B_2, \ldots, B_n$ are corresponding formulae defined by Definition 9, then:

$$B_1, B_2, \ldots, B_n \vdash A'$$

Proof. (*By induction*). We apply induction on the degree of A, i.e., the number n of logical connectives in A.

(*Base case*). $n = 0$.
In the case that $n = 0$, A is an atomic sentence (i.e. it has no logical connectives) and thus consists of only a single propositional variable, say a. We then have two cases to consider: either $v^*(A) = true$ or $v^*(A) = false$.

Case 1: $v^*(A) = true$
Clearly, based on Definition 9, this implies that $A' = A = a$ and $B_1 = a$. Substituting these into the lemma, we have to prove that $a \vdash a$.
Since we know that every proposition implies itself, then we have $\vdash a \to a$ since this is tautological. By applying Deduction Theorem to this, we then have $a \vdash a$.
Case 2: $v^*(A) = false$
This implies that $A' = \neg A = \neg a$ and $B_1 = \neg a$. Substituting these into the lemma, we have to prove that $\neg a \vdash \neg a$.
By the same reasoning as above, we have $\vdash \neg a \to \neg a$. By applying Deduction Theorem to this, we then have $\neg a \vdash \neg a$.

(*Inductive step*). Assume that the lemma holds for any A with degree $j < n$. We now want to prove that it holds for A with degree n . There are several subcases to deal with:
Case 1: A is $\neg A_1$
In this case, A_1 has less than n connectives. By the inductive assumption, we have the formulae $A_1', B_1, B_2, \ldots, B_n$ corresponding to A_1 and the propositional variables $b_1, b_2, \ldots, b_n$ in A_1, as defined by Definition 9, such that $B_1, B_2, \ldots, B_n \vdash A_1'$. We can observe that the formulae A and $\neg A_1$ have the same propositional variables, so the corresponding formulae $B_1, B_2, \ldots, B_n$ for each are also the same for both of them. With our inductive assumption above, $B_1, B_2, \ldots, B_n \vdash A_1'$, we are now going to prove that $B_1, B_2, \ldots, B_n \vdash A'$. There are two cases to consider:

Case 1a): $v^*(A) = true$
If this is the case, then we know by Definition 9 that $A_1' = A_1$. By the inductive assumption $B_1, B_2, \ldots, B_n \vdash A_1'$, we now obtain $B_1, B_2, \ldots, B_n \vdash A_1$. In this case, $v^*(A) = v^*(\neg A_1) = \neg v^*(true) = false$, and so $A' = \neg A = \neg\neg A_1$. Since we know that it is tautological that $\vdash (A_1 \to \neg\neg A_1)$, we have by monotonicity that $B_1, B_2, \ldots, B_n \vdash (A_1 \to \neg\neg A_1)$. By $B_1, B_2, \ldots, B_n \vdash A_1$ that we obtained previously and modus ponens, we have $B_1, B_2, \ldots, B_n \vdash \neg\neg A_1$. In other words, this is equivalent to $B_1, B_2, \ldots, B_n \vdash \neg A$, which is also equivalent to $B_1, B_2, \ldots, B_n \vdash A'$.

Case 1b): $v^*(A) = false$
If this is the case, then we know by Definition 9 that $A_1' = \neg A_1$. Thus, $v^*(A) = true$ so $A' = A$. Thus, from the inductive assumption above, we then have $B_1, B_2, \ldots, B_n \vdash \neg A_1$, which means then we have $B_1, B_2, \ldots, B_n \vdash A'$.

Case 2: A is $(A_1 \to A_2)$
In this case, $A = A(b_1, \ldots, b_n)$, so there are some subsequences $c_1, \ldots, c_k$ and $d_1, \ldots, d_m$ (where $k, m \leq n$) of the sequence $b_1, \ldots, b_n$ such that $A_1 = A(c_1, \ldots, c_k)$ and $A_2 = A(d_1, \ldots, d_m)$. We know that A_1 and A_2 have less than n connectives, so by the inductive formula, we will obtain the appropriate formulae $C_1, \ldots, C_k$ and $D_1, \ldots, D_m$ such that $C_1, C_2, \ldots, C_k \vdash A_1'$ and $D_1, D_2, \ldots, D_m \vdash A_2'$, where $C_1, C_2, \ldots, C_k$ and $D_1, D_2, \ldots, D_m$ are subsequences of the formulae $B_1, B_2, \ldots, B_n$ corresponding to the propositional variables in A. By monotonicity, we then have $B_1, B_2, \ldots, B_n \vdash A_1'$ and $B_1, B_2, \ldots, B_n \vdash A_2'$. There are three cases to consider:

Case 2a): $v^*(A_1) = v^*(A_2) = true$
If $v^*(A_1) = true$ then A_1' is A_1, and if $v^*(A_2) = true$ then A_2' is A_2. We also have $v^*(A_1 \to A_2) = true$ and so A' is $(A_1 \to A_2)$. By the above and $B_1, B_2, \ldots, B_n \vdash A_2'$, we then have $B_1, B_2, \ldots, B_n \vdash A_2$. Then, by Axiom 1, we also have $\vdash (A_2 \to (A_1 \to A_2))$.

Then, by monotonicity and applying modus ponens, we derive $B_1, B_2, \ldots, B_n \vdash (A_1 \to A_2)$, which means $B_1, B_2, \ldots, B_n \vdash A'$

Case 2b): $v^*(A_1) = true, v^*(A_2) = false$,
If $v^*(A_1) = true$ then A_1' is A_1, and if $v^*(A_2) = false$ then A_2' is $\neg A_2$. In this case, we also have $v^*(A_1 \to A_2) = false$ and so A' is $\neg(A_1 \to A_2)$. By the above and Definition 9, we have $B_1, B_2, \ldots, B_n \vdash A_1$ and $B_1, B_2, \ldots, B_n \vdash \neg A_2$. Since we know that $\vdash (A_1 \to (\neg A_2 \to \neg(A_1 \to A_2)))$ because it is tautological, we know by monotonicity and applying modus ponens twice that $B_1, B_2, \ldots, B_n \vdash \neg(A_1 \to A_2)$, which means $B_1, B_2, \ldots, B_n \vdash A'$.

Case 2c): $v^*(A_1) = false$
If this is the case, then A_1' is $\neg A_1$, and, regardless of whatever truth value v assigns to A_2, we have $v^*(A_1 \to A_2) = true$, and so A' is $(A_1 \to A_2)$. Therefore, by Definition 9 and above, we have $B_1, B_2, \ldots, B_n \vdash \neg A_1$. We also know tautologically that $\vdash (\neg A_1 \to (A_1 \to A_2))$. By monotonicity and applying modus ponens, we have $B_1, B_2, \ldots, B_n \vdash (A_1 \to A_2)$, which means $B_1, B_2, \ldots, B_n \vdash A'$.

This means that we have covered all possible cases. Therefore, by induction on n, the proof of the lemma is complete. ∎

To continue the proof of the theorem, first we assume that $\Gamma \vDash A$. Let $b_1, b_2, \ldots, b_n$ be all propositional variables that occur in A, i.e., $A = A(b_1, b_2, \ldots, b_n)$.
Let v : propositional variables $\to$ {truth, false} be any variable of truth assignment,
and let $v_A : \{b_1, b_2, \ldots, b_n\} \to$ {truth, false} be its restriction to the formula A,
i.e., $v_A = v \mid \{b_1, b_2, \ldots, b_n\}$.
Let $V_A = \{v_A : v_A : \{b_1, b_2, \ldots, b_n\} \to \{\text{truth, false}\}\}$.
By Lemma 10 and the assumption that $\Gamma \vDash A$, any $v \in V_A$ defines formulae $B_1, B_2, \ldots, B_n$ such that $B_1, B_2, \ldots, B_n \vdash A$.
The proof is based on a method of using all of $v \in V_A$ to define a process of elimination of all hypothesis $B_1, B_2, \ldots, B_n$ in the above equation, $B_1, B_2, \ldots, B_n \vdash A$, to finally construct the proof of A in Γ, i.e., to prove that $\Gamma \vdash A$.

Step 1: Elimination of B_n
We observe by Definition 9 that each B_i is b_i or $\neg b_i$ depending on the choice of $v \in V_A$. In particular, $B_n = b_n$ or $B_n = \neg b_n$. We then choose two truth assignments $v_1 \neq v_2 \in V_A$ such that $v_1 \mid \{b_1, \ldots, b_{n-1}\} = v_2 \mid \{b_1, \ldots, b_{n-1}\}$ and $v_1(b_n) = true$ and $v_2(b_n) = false$.

Case 1: $v_1(b_n) = true$
Thus, by Definition 9 we have $B_n = b_n$. By $v_1 \mid \{b_1, \ldots, b_{n-1}\} = v_2 \mid \{b_1, \ldots, b_{n-1}\}$, assumption that $\Gamma \vDash A$ and Lemma 10 applied to v_1, we have $B_1, B_2, \ldots, B_{n-1}, b_n \vdash A$. By Deduction Theorem, we then have $B_1, B_2, \ldots, B_{n-1} \vdash (b_n \to A)$.

Case 1: $v_1(b_n) = false$
Thus, by Definition 9 we have $B_n = \neg b_n$. By $v_1 \mid \{b_1, \ldots, b_{n-1}\} = v_2 \mid \{b_1, \ldots, b_{n-1}\}$, assumption that $\Gamma \vDash A$ and Lemma 10 applied to v_2, we have $B_1, B_2, \ldots, B_{n-1}, \neg b_n \vdash A$.

By Deduction Theorem, we then have $B_1, B_2, \ldots, B_{n-1} \vdash (\neg b_n \to A)$. We know tautologically that $(\phi \to \psi) \to ((\neg\phi \to \psi) \to \psi)$, thus we have that $\vdash (b_n \to A) \to ((\neg b_n \to A) \to A)$. By monotonicity, we have that $B_1, B_2, \ldots, B_{n-1} \vdash (b_n \to A) \to ((\neg b_n \to A) \to A)$. Then, we apply modus ponens twice to $B_1, B_2, \ldots, B_{n-1} \vdash (b_n \to A) \to ((\neg b_n \to A) \to A)$ and $B_1, B_2, \ldots, B_{n-1} \vdash (\neg b_n \to A)$, we then derive $B_1, B_2, \ldots, B_{n-1} \vdash A$.

Thus, we have eliminated B_n.

Step 2: Elimination of B_{n-1} from $B_1, B_2, \ldots, B_{n-1} \vdash A$

We repeat Step 1. As with before, we have two cases to consider: $B_{n-1} = b_{n-1}$ or $B_{n-1} = \neg b_{n-1}$. We then choose two truth assignments $w_1 \neq w_2 \in V_A$ such that $w_1 | \{b_1, \ldots, b_{n-2}\} = w_2 | \{b_1, \ldots, b_{n-2}\} = v_1 | \{b_1, \ldots, b_{n-2}\} = v_2 | \{b_1, \ldots, b_{n-2}\}$ and $w_1(b_{n-1}) = true$ and $w_2(b_{n-1}) = false$.
As with before, we apply Lemma 10, Deduction Theorem, monotonicity, the tautology $(\phi \to \psi) \to ((\neg\phi \to \psi) \to \psi)$ and modus ponens twice to eliminate B_{n-1} just as we have eliminated B_n.

After n steps, we finally obtain that $\vdash A$, i.e., $\Gamma \vdash A$. Therefore, this ends the proof of Completeness Theorem. ∎

2. General Mathematical Remarks

The formal system we are working on, the Hilbert proof system, is an incredibly important system in mathematical logic. Due to its minimalist nature, where it only usually includes one or two rules of inferences, it makes it a rigorous and axiomatic language to formalise much of mathematics. The soundness and completeness theorems are two of the most important theorems once can prove in a formal language because it tells us that the proof system is reliable and effective.
In a logic course, propositional logic is usually the first type of language one learns, usually known as *zeroth order logic*, before moving on to higher order logic like predicate logic and quantifier logic. Therefore, while more advanced systems of logic are more useful nowadays, understanding propositional logic is essential because a lot of the same reasoning is used for other results in these higher logics.
In particular, understanding the completeness theorem for propositional logic provides a firm grounding for understanding how to proof the completeness theorem for each of the higher level logics, to finally be able to prove and understand one of the most important theorems in mathematical logic, and arguably one of the most important theorems in the 20th century, which is Gödel's Incompleteness Theorem. In basic terms, it posits that a logical system that is powerful enough to represent certain basic mathematical properties will necessarily be incomplete. This means that there are true statements that cannot be proven by that system.

3. Lean Theorem Prover

Lean 3.0 is a programming language designed as a theorem prover for mathematics. In using it, most people rely on its online library that the community continues to expand called *mathlib* which includes a lot of basic mathematical tools, functions and theorems once it is imported. The language relies a lot on what is called *types* because it is a language known as dependent type theory. What

this means is that each mathematical object that we use in our code must be given a specific type, and in turn, every function that we use or define ourselves also only takes a certain type as input and outputs a certain type. This constant reliance on types means that one cannot make any unjustified *jumps* in reasoning like how one would on paper, and in turn, it means that every step of a proof in Lean must be fully justified, which gives it an increase in rigour.

4. Comments on Lean Implementation [to be reviewed and retyped after the code is finished]

- Key steps in math and how they were implemented:
- 1) Usage of string to have a collection of atomic sentences and logical connectors
- 2) Usage of lists to call a specific atomic sentence in a formula that is composed of multiple propositional variables
- 3)

5. Future Improvements

- Have a cleaner *single* definition of atomic sentences and formulae, figure out whether want to make it set of propositions or string of booleans, etc

6. Appendix (Lean Code)