0 Conventions

The natural numbers include 0, so $\mathbb{N} = \{0, 1, 2, \dots\}$.

This document is written in the **metalanguage** of mathematical English. We assume familiarity with the basic concepts of set theory and **metalogical** reasoning. In this context, **metalogic** is that thing which, in the metalanguage, is usually called "logic." On the other hand, the subject of this document is **propositional logic** (a type of *formal logic*). In Section 1, we describe the **object langauge** of propositional logic. In later sections, we go on to describe the other components of propositional logic.

Many of the concepts we describe about the object language have parallel concepts in the metalanguage. Often, there is a particular metalogical concept which must be encoded in the object language. To do this, we might invoke the metalogical concept to explain or justify the corresponding formal logic concept. Best efforts are made to keep the distinction between the two clear.

1 Language

Definition 1.1. A **propositional logic language** L consists of the following data:

(a) a nonempty set of symbols called **propositional variables** (e.g., $P, Q, R, \dots$),

(b) a set of **logical connectives** (e.g., $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$).

Alternate names for propositional logic include the following: propositional calculus, sentential logic (operating on sentences), statement logic, and zeroth-order logic.

Remark 1.2. In practice, the propositional variables are either be upper or lower case letters. If we run out of letters, we can use numerical subscripts (e.g., $P_1, P_2, \dots$).

Remark 1.3. For historical reasons, the following five logical connectives are considered canonical in some sense:

Symbol	Formal Name	Informal Name	Alternate Symbols
$\neg$	negation	"not"	$\sim$, !, $'$, $-$
$\wedge$	conjunction	"and"	&, &&, $\cdot$
$\vee$	disjunction	"or"	$\vert$, $\Vert$, $+$
$\rightarrow$	implication	"implies" or "if-then"	$\implies$, $\supset$
$\leftrightarrow$	biconditional	"if and only if" or "iff"	$\iff$, $\equiv$

The alternate symbols may be found in older texts or for logic in other contexts. We will only use the symbols on the left of the above table to refer to the logical connectives proper. Any usage of a symbol on the right means something else. For example, " $\implies$ " and " $\iff$ " are used in their normal manner of mathematical English.

There are other logical connectives that are used in other contexts (e.g., XOR, NAND, NOR), but we will see later that they can be defined in terms of the five canonical connectives, so they are not strictly necessary. In fact, we don't even need all five of the canonical connectives, as we can define some in terms of others. Some equivalencies are sketched below and will be treated more formally later.

$$(P \wedge Q) \approx (\neg(\neg P \vee \neg Q)) \qquad (P \vee Q) \approx (\neg(\neg P \wedge \neg Q))$$
$$(P \to Q) \approx (\neg P \vee Q) \qquad (P \leftrightarrow Q) \approx ((P \to Q) \wedge (Q \to P))$$

For our purposes, we will take either $\{\neg, \vee\}$ or $\{\neg, \to\}$ as the minimal set of logical connectives.

Remark 1.4. Sometimes **logical constants** are included in the language, e.g., $\top$ and $\bot$ for truth/tautology and falsehood/contradiction, respectively. We could call nullary connectives, in the same sense that negation is a unary connective and the others are binary connectives. More generally, we could consider n-ary connectives for $n \geq 0$. In which case, $\{\bot, \to\}$ is also a possible minimal set of logical connectives.

Remark 1.5. Sometimes **auxiliary symbols** are included in the language to make formulas easier to read, e.g., parentheses, brackets, and commas. We will use parentheses here to disambiguate the order of operations. Something like reverse Polish notation can achieve the same effect without parentheses, but at the cost of readability.

Definition 1.6. Let L be a language of propositional logic. A **formula** of L, also called a **well-formed formula** or **wff**, is defined inductively as follows:

(a) Every propositional variable is a formula.

(b) If φ is a formula, then $(\neg\varphi)$ is a formula.

(c) If φ and ψ are formulas, then $(\varphi \star \psi)$ is a formula, where $\star$ is any binary logical connective, e.g., $\vee$ or $\to$.

Any sequence of symbols that is not obtained by one of the above rules is not a formula.

Remark 1.7. Implicitly, we are allowing the use of parentheses to disambiguate the order of operations. A proper treatment of this aspect of formal languages can be found elsewhere.

Remark 1.8. One way to characterize the set of formulas is as the smallest set of strings of symbols which contains all the propositional variables and is closed under finite applications of the logical connective operations. Let $\mathcal{V} = \mathcal{V}(L)$ be the set of propositional variables of L. Then the set $\mathcal{F} = \mathcal{F}(L)$ of formulas of L is the set generated by $\mathcal{V}$ under the following **formula building operations**:

$$\begin{aligned}
\varepsilon_\neg(\varphi) &:= (\neg\varphi), \\
\varepsilon_\wedge(\varphi, \psi) &:= (\varphi \wedge \psi), \\
\varepsilon_\vee(\varphi, \psi) &:= (\varphi \vee \psi), \\
\varepsilon_\to(\varphi, \psi) &:= (\varphi \to \psi), \\
\varepsilon_\leftrightarrow(\varphi, \psi) &:= (\varphi \leftrightarrow \psi).
\end{aligned}$$

In other words, $\mathcal{F} = \overline{\mathcal{V}}$ is the closure of $\mathcal{V}$ under these operations.

Definition 1.9. Let $\varphi \in \mathcal{F}$ be a formula, $A \in \mathcal{V}(\varphi)$ a propositional variable occurring in φ, and $B \in \mathcal{V}$ any propositional variable. Then there is a formula obtained from φ via **substitution** *of* B *for* A, written as $\varphi[B/A]$ or $\varphi_A[B]$. This operation is defined on formulas recursively as follows:

$$\begin{aligned} A[B/A] &:= B, \\ C[B/A] &:= C \quad \text{for} \quad C \in \mathcal{V} \setminus \{A\}, \\ (\neg\varphi)[B/A] &:= \neg(\varphi[B/A]), \\ (\varphi \star \psi)[B/A] &:= (\varphi[B/A] \star \psi[B/A]) \quad \text{for} \quad \star \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}. \end{aligned}$$

2 Semantics

Definition 2.1. Let L be a language of propositional logic. A **truth valuation** on L is a map $v : \mathcal{V} \rightarrow \{T, F\}$ from the set of propositional variables to the set of truth values.

Remark 2.2. A truth valuation may also be called a truth assignment or structure or interpretation of L. All of these terms express the fact that v provides **semantics** for L. In the context of logic, semantics refers to something like the meaning of the words. Semantics for a language tell us what the words and sentences refer to. etc

Remark 2.3. A truth valuation v can be *uniquely* extended to a map $\bar{v} : \mathcal{F} \rightarrow \{T, F\}$ by defining it recursively on formulas of L:

$$\begin{aligned} \bar{v}(A) &:= v(A) \quad \text{for} \quad A \in \mathcal{V}, \\ \bar{v}(\neg\varphi) &:= \begin{cases} T & \text{if } \bar{v}(\varphi) = F, \\ F & \text{otherwise,} \end{cases} \\ \bar{v}(\varphi \wedge \psi) &:= \begin{cases} T & \text{if } \bar{v}(\varphi) = T \text{ and } \bar{v}(\psi) = T, \\ F & \text{otherwise,} \end{cases} \\ \bar{v}(\varphi \vee \psi) &:= \begin{cases} T & \text{if } \bar{v}(\varphi) = T \text{ or } \bar{v}(\psi) = T, \\ F & \text{otherwise,} \end{cases} \\ \bar{v}(\varphi \rightarrow \psi) &:= \begin{cases} T & \text{if } \bar{v}(\varphi) = F \text{ or } \bar{v}(\psi) = T, \\ F & \text{otherwise,} \end{cases} \\ \bar{v}(\varphi \leftrightarrow \psi) &:= \begin{cases} T & \text{if } \bar{v}(\varphi) = \bar{v}(\psi), \\ F & \text{otherwise.} \end{cases} \end{aligned}$$

Displayed below in a truth table format, where the rows correspond to the possible truth values of the relevant propositional variables:

φ	ψ	$\neg\varphi$	$\varphi \wedge \psi$	$\varphi \vee \psi$	$\varphi \to \psi$	$\varphi \leftrightarrow \psi$
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

These definitions are chosen in such a way that they mirror the meanings of these concepts as they appear in our metalanguage of mathematical English.

It should be noted that both the existence and uniqueness of the extension $\bar{\nu}$ are worth proving. This fact depends on some of the specifics of the language L, but it should come about easily if formulas can be uniquely parsed into abstract syntax trees. Taking uniqueness for granted, we will only refer to ν, identifying it with its extension $\bar{\nu}$.

In this manner, we consider a truth valuation to be a map $\nu : \mathcal{F} \to \{T, F\}$ which is characterized by the restriction $\nu|_{\mathcal{V}} : V \to \{T, F\}$ to propositional variables.

Definition 2.4. Let L be a language of propositional logic, ν a truth valuation on L.

For a formula $\varphi \in \mathcal{F}$, we say that ν **satisfies** φ if $\nu(\varphi) = T$. Equivalently, we might say φ is **true** in ν when $\nu(\varphi) = T$, and **false** if $\nu(\varphi) = F$.

For a set of formulas $\Sigma \subseteq \mathcal{F}$, say ν **satisfies** Σ if ν satisfies all the formulas in Σ.

A formula or set of formulas is **satisfiable** if there exists a truth valuation satisfying it.

Definition 2.5. Σ **tautologically implies** τ (written $\Sigma \vDash \tau$) if every truth valuation satisfying Σ also satisfies τ.

In case $\varnothing \vDash \tau$, say τ is a **tautology** and write $\vDash \tau$.

In case $\{\sigma\} \vDash \tau$, write $\sigma \vDash \tau$.

If $\varphi \vDash \psi$ and $\psi \vDash \varphi$, say φ and ψ are **tautologically equivalent** and write $\varphi \equiv \psi$.

Remark 2.6. When $\Sigma \vDash \tau$, it may also be said that Σ **semantically implies** τ, or that τ is a **tautological/semantic consequence** of Σ. Similarly, $\varphi \equiv \psi$ could be read as semantic equivalence.

Remark 2.7. Recall that truth valuations on L are characterized by their valuation on the set of variables $\mathcal{V} = \mathcal{V}(L)$. So there is a bijection between the set of truth valuations on L and the set of functions

$$\{T, F\}^{\mathcal{V}} = \{\text{functions } \mathcal{V} \to \{T, F\}\}.$$

This set has cardinality $2^{|\mathcal{V}|}$, hence there are as many truth valuations on L.

In particular, if $\mathcal{V}$ is a finite set then there are only finitely many truth valuations on L. In which case, it is "straightforward" to simply check all possible truth valuations to determine if $\Sigma \vDash \tau$ or indeed if $\vDash \tau$. However, the number of cases to check grows exponentially with the number of variables, so it quickly becomes practically intractable.

Note, however, that for any given formula $\varphi \in \mathcal{F}$, there are only finitely many propositional variables—say $\mathcal{V}(\varphi) = \{A_1, \dots, A_n\}$—which occur in φ. Informally, we should expect that the truth valuation of φ to only depend on the truth valuations of the A_i's. One way to formalize this claim is as follows: if two truth valuations agree on $\mathcal{V}(\varphi)$ then they must also agree on φ, i.e., if $v|_{\mathcal{V}(\varphi)} = v'|_{\mathcal{V}(\varphi)}$ then $v(\varphi) = v'(\varphi)$.

Fact 2.8. Let v and v' be two truth valuations on L, let $\varphi \in \mathcal{F}$ be a formula of L, and let $\mathcal{V}(\varphi)$ be the set of propositional variables occurring in φ. If $v|_{\mathcal{V}(\varphi)} = v'|_{\mathcal{V}(\varphi)}$ then $v(\varphi) = v'(\varphi)$.

In general

In practice, this means that we can determine whether or not $\Sigma \models \tau$, only needing to consider variables occurring in τ and formulas of Σ. To account for this, we could consider equivalence classes of truth valuations. Or, equivalently, we could modify our definition of truth valuations so that they are relative to a subset $S \subseteq \mathcal{V}$ of variables of L, instead of all variables.

Example 2.9. A selected list of tautologies (like most of these notes, lifted from Enderton)

1. Associativity and Commutativity.

$$\begin{array}{cc} (A \wedge (B \wedge C)) \leftrightarrow ((A \wedge B) \wedge C) & (A \wedge B) \leftrightarrow (B \wedge A) \\ (A \vee (B \vee C)) \leftrightarrow ((A \vee B) \vee C) & (A \vee B) \leftrightarrow (B \vee A) \\ (A \leftrightarrow (B \leftrightarrow C)) \leftrightarrow ((A \leftrightarrow B) \leftrightarrow C) & (A \leftrightarrow B) \leftrightarrow (B \leftrightarrow A) \end{array}$$

2. Distributivity.

$$(A \wedge (B \vee C)) \leftrightarrow ((A \wedge B) \vee (A \wedge C))$$
$$(A \vee (B \wedge C)) \leftrightarrow ((A \vee B) \wedge (A \vee C))$$

3. Negation.

$$(\neg(\neg A)) \leftrightarrow A$$
$$(\neg(A \to B)) \leftrightarrow (A \wedge (\neg B))$$
$$(\neg(A \leftrightarrow B)) \leftrightarrow ((A \wedge (\neg B)) \vee ((\neg A) \wedge B))$$

4. De Morgan's Laws.

$$(\neg(A \wedge B)) \leftrightarrow ((\neg A) \vee (\neg B))$$
$$(\neg(A \vee B)) \leftrightarrow ((\neg A) \wedge (\neg B))$$

5. Other

Excluded Middle	$A \vee (\neg A)$
Non-Contradiction	$\neg(A \wedge (\neg A))$
Contraposition	$(A \to B) \leftrightarrow ((\neg B) \to (\neg A))$
Exportation	$((A \wedge B) \to C) \leftrightarrow (A \to (B \to C))$

Claim 2.10. For $\Sigma \subseteq \mathcal{F}$ and $\alpha, \beta \in \mathcal{F}$,

(i) $\Sigma \cup \{\alpha\} \vDash \beta$ iff $\Sigma \vDash (\alpha \to \beta)$,

(ii) $\alpha \equiv \beta$ iff $\vDash (\alpha \leftrightarrow \beta)$.

3 Compactness

Theorem 3.1 (Compactness for Propositional Logic)**.** A set of formulas is satisfiable iff every finite subset is satisfiable.

Proof. The ($\Longrightarrow$) direction of the proof is clear: if ν satisfies Σ then it also satisfies every finite subset of Σ.

For the ($\Longleftarrow$) direction, let $\Sigma \subseteq \mathcal{F}$ be a (possibly infinite) set of formulas. Moreover, assume Σ is **finitely satisfiable**, i.e., every finite subset of Σ is satisfiable.

We extend Σ to a maximal set of formulas Δ using Zorn's Lemma. (cf. Enderton p59. There, compactness is proven for a countable language without the Axiom of Choice, using an enumeration of the formulas.) Consider the set

$$\mathcal{S} := \{U \mid \Sigma \subseteq U \subseteq \mathcal{F} \text{ and } U \text{ is finitely satisfiable}\},$$

ordered with the usual subset relation $\subseteq$. Let $C \subseteq \mathcal{S}$ be a chain (a totally ordered subset), and denote the union

$$\mathcal{U} := \bigcup_{U \in C} U.$$

Clearly, $\mathcal{U}$ is an upper bound of C. Moreover, $\Sigma \subseteq \mathcal{U} \subseteq \mathcal{F}$ and we claim that $\mathcal{U}$ is finitely satisfiable, whence $\mathcal{U} \in \mathcal{S}$. Let $\{\alpha_1, \ldots, \alpha_n\} \subseteq \mathcal{U}$ be a finite subset of formulas. By construction of $\mathcal{U}$, each α_i is an element of some $U_i \in C$. Since C is totally ordered with respect to the subset relation, there is a maximum $U = \max_{\subseteq}\{U_1, \ldots, U_n\}$. Then $\{\alpha_1, \ldots, \alpha_n\} \subseteq U$ and U is finitely satisfiable, so $\{\alpha_1, \ldots, \alpha_n\}$ is satisfiable. Thus, $\mathcal{U}$ is finitely satisfiable and is therefore an upper bound of C in $\mathcal{S}$.

By Zorn's Lemma, a maximal $\Delta \in \mathcal{S}$ exists. Automatically, we know $\Sigma \subseteq \Delta$ and Δ is finitely satisfiable.

Claim 3.2. For all $\alpha \in \mathcal{F}$, either $\alpha \in \Delta$ or $(\neg\alpha) \in \Delta$.

Proof of Claim. Assume for contradiction that the claim is false, i.e., there exists a formula α such that $\alpha \notin \Delta$ and $(\neg\alpha) \notin \Delta$.

We will show that if this is the case, then Δ can be extended by either α or $\neg\alpha$ while remaining finitely satisfiable.

By assumption, $\Delta \cup \{\alpha\}$ is not finitely satisfiable. In other words, there exists a finite, unsatisfiable subset $\{\beta_1, \ldots, \beta_n, \alpha\}$. We know α is included since just the β_i's would be satisfiable as a finite subset of Δ. Similarly, $\Delta \cup \{\neg\alpha\}$ is not finitely satisfiable, so there exists a finite, unsatisfiable subset $\{\gamma_1, \ldots, \gamma_m, \neg\alpha\}$.

Now consider the finite subset

$$S := \{\beta_1, \ldots, \beta_n, \gamma_1, \ldots, \gamma_m\} \subseteq \Delta.$$

Then S is satisfiable; let v be a truth valuation satisfying it. Then either $v(\alpha) = \mathsf{T}$ or $v(\neg\alpha) = \mathsf{T}$. The former implies that $\{\beta_1, \ldots, \beta_n, \alpha\}$ is satisfiable, while the latter implies that $\{\gamma_1, \ldots, \gamma_m, \neg\alpha\}$ is satisfiable. Either case is a contradiction, so the claim holds. □

We now wish to construct a truth valuation v satisfying Δ. For propositional variables $A \in \mathcal{V}$, put

$$v(A) := \mathsf{T} \quad \text{iff} \quad A \in \Delta.$$

Of course, this characterizes v on all of $\mathcal{F}$, but it remains to show that v satisfies Δ. In fact, not only does v satisfy Δ, but we can say something slightly stronger.

Claim 3.3. For all $\alpha \in \mathcal{F}$, v satisfies α iff $\alpha \in \Delta$.

We first perform a construction that will aid in the proof of this claim. Let $\alpha \in \mathcal{F}$ be any formula. Additionally, let $\mathcal{V}(\alpha)$ be the (necessarily finite) set of propositional variables occurring in α. Moreover, define the set's elementwise negation

$$\neg\mathcal{V}(\alpha) := \{\neg A \mid A \in \mathcal{V}(\alpha)\}$$

to be the set of negations of the propositional variables occurring in α. Then the set

$$V := (\mathcal{V}(\alpha) \cup \neg\mathcal{V}(\alpha)) \cap \Delta$$

is the set of propositional variables occurring in α and their negations which are in Δ. This set has two important properties for our purposes. First, the truth of a valuation on α is completely determined by its truth on the variables in V. This follows from Claim 3.2—for every propositional variable $A \in \mathcal{V}(\alpha)$, either A or its negation $\neg A$ is an element of Δ and therefore an element of V. Second, v satisfies V, since $V \subseteq \Delta$.

Assume $\alpha \in \Delta$. Now $V \cup \{\alpha\}$ is a finite subset of Δ and is therefore satisfiable; let v' be a truth valuation satisfying it. Once again, v and v' agree on V, so they must also agree on α. Hence $v(\alpha) = v'(\alpha) = \mathsf{T}$.

Assume $(\neg\alpha) \in \Delta$. Since $V \cup \{\neg\alpha\}$ is a finite subset of Δ, there exists a truth valuation v' satisfying it. But this means v and v' agree on V, so they must agree on α. And since $v'(\neg\alpha) = \mathsf{T}$, we must have $v(\alpha) = v'(\alpha) = \mathsf{F}$.

This completes the proof of Claim 3.3, hence compactness. □

Corollary 3.4. If $\Sigma \models \tau$ then there exists a finite $\Sigma_0 \subseteq \Sigma$ such that $\Sigma_0 \models \tau$.

(Are we assuming Σ is satisfiable?)

Proof. Note that $\Sigma \vDash \tau$ iff $\Sigma \cup \{\neg\tau\}$ is unsatisfiable. worth proving?

We prove by contrapositive.

Assume there does not exist a finite $\Sigma_0 \subseteq \Sigma$ which tautologically implies τ. Then for every finite $\Sigma_0 \subseteq \Sigma$, the set $\Sigma_0 \cup \{\neg\tau\}$ is satisfiable. Equivalently, for every every finite $\Sigma_0 \subseteq \Sigma$, the set $\Sigma_0 \cup \{\neg\tau\}$ is satisfiable. It follows that $\Sigma \cup \{\neg\tau\}$ is finitely satisfiable, and compactness implies it is satisfiable. By the fact, $\Sigma \nvDash \tau$. □

Remark 3.5. This Corollary is equivalent to the compactness theorem. One can show that compactness is provable from the Corollary. In fact, such a proof is easier than the proof of compactness itself.

4 Deduction

Definition 4.1. An **inference rule** consists of the following data:

(i) a finite set of formulas $\alpha_1, \ldots, \alpha_n \in \mathcal{F}$ called the *premises*,

(ii) a formula $\beta \in \mathcal{F}$ called the *conclusion*.

Such an inference rule is denoted by

$$\frac{\alpha_1, \ldots, \alpha_n}{\beta}.$$

Remark 4.2. Consider the metalogical inference rule of *modus ponens*, which typically takes the following form:

(1) If $\mathfrak{P}$, then $\mathfrak{Q}$.

(2) $\mathfrak{P}$.

(3) Therefore, $\mathfrak{Q}$.

In mathematical English, modus ponens tells us that whenever we know (1) and (2), we may deduce (3). To implement modus ponens as an inference rule of propositional logic, we may try taking as premises $P \to Q$ and P, and as conclusion Q. In which case, we would write

$$\frac{P \to Q, P}{Q}.$$

However, this only expresses modus ponens for the propositional variables P and Q. What if we want modus ponens for the propositional variables A and B? Or, indeed, for any formulas α and β?

Definition 4.3. An **inference rule schema** R consists of the following data:

(i) a finite set of metalogical variables/symbols $\mathfrak{X} = \{\mathfrak{x}_1, \ldots, \mathfrak{x}_k\}$,

(ii) finitely many metaformulas $\mathfrak{a}_1, \ldots, \mathfrak{a}_n \in \mathcal{F}_{\mathfrak{X}}$,

(iii) a metaformula $\mathfrak{b} \in \mathcal{F}_{\mathfrak{X}}$,

where $\mathcal{F}_{\mathfrak{X}}$ is the closure of $\mathfrak{X}$ under the formula building operations, i.e., the language L with only the $\mathfrak{x}_i$'s as the variables.

An **instance** of R is an inference rule obtained by substituting formulas for the $\mathfrak{x}_i$'s in the metaformulas. Given formulas $\chi_1, \ldots, \chi_k \in \mathcal{F}$, the instance $R[\chi_1, \ldots, \chi_k]$ of R defined by formulas has

(i) premises $\alpha_i = \mathfrak{a}_i[\chi_1/\mathfrak{x}_1, \ldots, \chi_k/\mathfrak{x}_k]$ for $i = 1, \ldots, n$,

(ii) conclusion $\beta = \mathfrak{b}[\chi_1/\mathfrak{x}_1, \ldots, \chi_k/\mathfrak{x}_k]$.

Remark 4.4. An inference rule schema provides a way of describing a family of inference rules which share a common structure. In practice,

Definition 4.5. A **deductive system** or **proof system** (Π?) *for propositional logic* consists of the following data:

(i) a set Λ of tautologies called **logical axioms**,

(ii) a set of **inference rules** or **rules of deduction**,

Remark 4.6. Some common axioms are listed below. they are usually taken to be some tautologies

Name	Abbr.	Tautology
Law of the Excluded Middle	LEM	$A \vee (\neg A)$
Law of Non-Contradiction	LNC	$\neg(A \wedge (\neg A))$
P2	P2(i)	$(A \rightarrow (B \rightarrow A))$
	P2(ii)	$((A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)))$
	P2(iii)	$((\neg B \rightarrow \neg A) \rightarrow (A \rightarrow B))$

Table 1: Common Tautologies in Propositional Logic

Remark 4.7. Some common inference rules are listed in Table 2 below.

Table 2: Common Inference Rules of Propositional Logic

Name	Abbr.	Premise(s)	Conclusion(s)
Modus Ponens	MP	$\alpha \rightarrow \beta, \alpha$	β
Modus Tollens	MT	$\alpha \rightarrow \beta, \neg\beta$	$\neg\alpha$
Conjunction Introduction or Adjunction	$\wedge$I or Adj	α, β	$\alpha \wedge \beta$

Continued on next page

Table 2: Common Inference Rules of Propositional Logic (Continued)

Conjunction Elimination or Simplification	$\wedge$E or Simp	$\alpha \wedge \beta$	α or β
Conjunction Associativity	$\wedge$-Ass	$\alpha \wedge (\beta \wedge \gamma)$ $(\alpha \wedge \beta) \wedge \gamma$	$(\alpha \wedge \beta) \wedge \gamma$ $\alpha \wedge (\beta \wedge \gamma)$
Conjunction Commutativity	$\wedge$-Comm	$\alpha \wedge \beta$	$\beta \wedge \alpha$
Contraction	Contr	$\alpha \vee \alpha$	α
Disjunction Introduction or Expansion	$\vee$I	α	$\alpha \vee \beta$ or $\beta \vee \alpha$
Disjunction Elimination or Disjunctive Syllogism	$\vee$E	$\alpha \vee \beta, \neg\alpha$	β
Disjunction Associativity	$\vee$-Ass	$\alpha \vee (\beta \vee \gamma)$ $(\alpha \vee \beta) \vee \gamma$	$(\alpha \vee \beta) \vee \gamma$ $\alpha \vee (\beta \vee \gamma)$
Disjunction Commutativity	$\vee$-Comm	$\alpha \vee \beta$	$\beta \vee \alpha$
Cut Rule	Cut	$\alpha \vee \beta, \neg\alpha \vee \gamma$	$\beta \vee \gamma$
Biconditional Introduction	$\leftrightarrow$I	$\alpha \rightarrow \beta, \beta \rightarrow \alpha$	$\alpha \leftrightarrow \beta$
Biconditional Elimination	$\leftrightarrow$E	$\alpha \leftrightarrow \beta, \alpha$	β

Remark 4.8 (Crazy Waffling)**.** Notice that axioms and inference rules are in some respects made of the same stuff. They play slightly different roles on our formalization of proofs, but both serve to encode properties we want to assert about out methods of proof.

Our proof system should encode the shape of arguments. We want to know that whenever an argument takes a certain form, then it is "correct". The axioms and inference rules are the most fundamental forms of arguments we can make. They are the building blocks of our proofs.

These basic rules are built out of simple tautologies that should generally "feel true."

Modus Ponens is perhaps the most fundamental rule of all. It states that if we *have* $\alpha \rightarrow \beta$ and α, then we should also *have* β. The term *have* here is intentionally vague to not overlap with any of the formal terminology. Ideally, this notion of have should overlap with truth and knowledge and whatever other semantics you want to attach to the language. The actual assertion of Modus Ponens should feel true: if it is the case that having α leads to having β, and we do indeed have α, then it should also be the case that we have β.

Applications of MP can appear circular in some respects. Once more, MP states that whenever we have $\alpha \rightarrow \beta$ and α, then we should also have β. And supposing we have obtained $\alpha \rightarrow \beta$ and α by some other means, then MP allows us to obtain β. However, notice that this application of MP in the language of propositional logic, also took the form of MP in the metalanguage of mathematical English. That is, from the following two facts:

(I) Modus Ponens: if we have $\{\alpha \to \beta, \alpha\}$, then we also have β,

(II) we have $\{\alpha \to \beta, \alpha\}$;

we can conclude that (III) we also have β. More specifically, (I) is structurally similar to $\alpha \to \beta$, (II) is structurally similar to α, and of course the final conclusion of (III) is structurally similar to β. In order to apply MP in the object language, we have packaged up the the object language facts into metalanguage facts and then applied MP in the metalanguage.

$$\text{have}\,\{\{\alpha \to \beta, \alpha\} \vdash \beta, \{\alpha \to \beta, \alpha\}\} \implies \text{have}\,\beta$$

One could look at the previous previous paragraph and once again notices that an implicit application of MP has occurred in the metalanguage. That is, in describing MP in the object language, we turn to the metalanguage, but to describe MP in the metalanguage, we have only the metalanguage itself. There is no meta-metalanguage upon which we can base our description of MP in the metalanguage.

Maybe we can take something like the following as a sort of core principle of logic, expressed in the most general terms possible:

$$\{\mathfrak{P} \rightsquigarrow \mathfrak{Q} \quad \text{and} \quad \mathfrak{P}\} \qquad \rightsquigarrow \qquad \mathfrak{Q}.$$

The symbols $\mathfrak{P}$ and $\mathfrak{Q}$ (mathfrak P and Q) stand for something like a statement or proposition or sentence or any other *assertion* of fact or truth or knowledge or belief or possibility or any other sort of notion that is of interest in regards to argumentation. Similarly, the arrow $\rightsquigarrow$ stands for something like implication or entailment or deduction or inference or construction or determination or any other sort of relation that is relevant to objects such as $\mathfrak{P}$ and $\mathfrak{Q}$. The expression $\mathfrak{P} \rightsquigarrow \mathfrak{Q}$ should be understood as the following assertion: that assertions of $\mathfrak{P}$ lead to assertions of $\mathfrak{Q}$. If, in addition, we assert $\mathfrak{P}$, then we should also assert $\mathfrak{Q}$.

In fact, we can see that MP is sort of a recursive definition of the symbol $\rightsquigarrow$. To characterize $\mathfrak{P} \rightsquigarrow \mathfrak{Q}$, we might ask what assertion should we put in the blank space to make the following assertion correct:

$$\{- \quad \text{and} \quad \mathfrak{P}\} \qquad \rightsquigarrow \qquad \mathfrak{Q}.$$

We might think of this as a sort of universal property of the squiggly arrow. To fill in this blank is to answer the following question: what assertion, when combined with an assertion of $\mathfrak{P}$, should lead to an assertion of $\mathfrak{Q}$? Clearly, $\mathfrak{P} \rightsquigarrow \mathfrak{Q}$ is the assertion we need. In a sense, the definition of $\mathfrak{P} \rightsquigarrow \mathfrak{Q}$ is that assertions of $\mathfrak{P}$ can be turned into assertions of $\mathfrak{Q}$.

Another way to see the redundancy is to look at a particular case. Returning to normal propositional logic, consider the formula

$$((P \to Q) \wedge P) \to Q,$$

which encodes the this MP structure. This formula is of course a tautology, which can be seen by examining its truth table:

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