CHAPTER TEN: PROOF AND SEMANTIC CONSEQUENCE

1 Introduction

It seems that natural language users are able to form judgments about the correctness of arguments, i.e. whether a certain utterance must be true if certain other utterances are assumed to be true. Intuitively, such judgments seem not to depend on contingent facts (how the world is), in particular whether the utterances in question are true or false. Rather they seem to depend solely on the *forms* of the sentences in question. The study of formal proofs (as opposed to the informal metalanguage (Mathese) proofs we have been doing so far) first arose as attempts to describe *deduction*, what we do when we make judgments about what follows from assumptions based solely on the forms of the sentences.

Just as we have been using PL as an extremely simplified model of natural language syntax (one which ignores all words except the "logic words"), so we can give a simple model of deduction in terms of a system of *axioms* and *rules* that characterize (at least in part) what formulas are deducible from what other formulas.

On an informal first pass toward making this more precise, let's write $\phi_1, \ldots, \phi_n \vdash \phi$ to express that, if we assume all the ϕ_i are true, then we can safely infer that ϕ must be true. Of course, if we have a way to ascertain that a formula is a necessary truth, then we should be able to deduce it from no assumptions at all: $\vdash T$.

Axioms are certain deductions, such as $\vdash T$ (the axiom of Truth) whose correctness seems so obvious that we simply take them as given at the outset. Likewise, it seems obvious that any formula can be correctly (albeit trivially) deduced if we have already assumed it to be true: $\phi \vdash \phi$. We call the collection of axioms of that form the axiom schema of Reflexivity.

Rules are "simple higher-level deductions", that is, minimal reasoning steps that let us conclude that if certain deductions (called *premisses*) are assumed correct, then so is some other deduction (called the *conclusion*). For example, the roles of conjunction ($\wedge$) and implication ($\rightarrow$) in deduction can be characterized by the following five rules (here Γ ranges over (possibly empty) finite sets of formulas), which are more carefully articulated forms of the generalizations about conjunction and implication stated in Chapter 9, sections 2 and 4[1]:

[1]For expository simplicity, the role of disjunction and negation in deduction is

Conjunction Elimination 1: if $\Gamma \vdash \phi \wedge \psi$, then $\Gamma \vdash \phi$

Conjunction Elimination 2: if $\Gamma \vdash \phi \wedge \psi$, then $\Gamma \vdash \psi$

Conjunction Introduction: if $\Gamma \vdash \phi$ and $\Gamma \vdash \psi$, then $\Gamma \vdash \phi \wedge \psi$

Implication Elimination: if $\Gamma \vdash \phi$ and $\Gamma \vdash \phi \rightarrow \psi$, then $\Gamma \vdash \psi$

Implication Introduction: if $\Gamma, \phi \vdash \psi$, then $\Gamma \vdash \phi \rightarrow \psi$

One last rule, which differs from those above in neither introducing nor eliminating a connective[2], is the following:

Weakening: if $\Gamma \vdash \psi$, then $\Gamma, \phi \vdash \psi$

The point of Weakening is that if a certain conclusion can be drawn from certain assumptions, then it can also be drawn from *more* assumptions. This may seem too obvious to even be worth stating, but before long we will encounter some formal proofs that depend on this principle. And in later chapters, we will consider some kinds of logic that disallow the use of Weakening.

The axioms and rules above are only a point of departure for constructing formal proofs. As we will see in the following section, the totality of formal proofs is defined by taking the axioms and rules as the base clauses and recursion clauses, respectively, in a recursive definition.

2 Provability in PIPL

Let Φ be the set of (PIPL) formulas over some given finite set of propositional letters. By a **context**, we mean a (possibly empty) finite set of formulas, and we denote by Con the set of contexts. We will now recursively define the relation $\vdash$ between contexts and formulas, variously called **provability**, **deducibility**, or **derivability**. To that end, we employ the following notational conventions:

We use ϕ, ψ, ξ, and ζ as metavariables over formulas.

We use Γ and Δ as metavariables over contexts.

postponed.

[2]Such rules are called *structural*.

A singleton context $\{\phi\}$ is abbreviated as ϕ, so $\phi \vdash \psi$ abbreviates $\{\phi\} \vdash \psi$.

The empty context is abbreviated by writing nothing, so $\vdash \psi$ abbreviates $\emptyset \vdash \psi$.

Contexts with more than one member are abbreviated by eliminating the set braces, so $\phi_1, \ldots, \phi_n \vdash \psi$ abbreviates $\{\phi_1, \ldots, \phi_n\} \vdash \psi$.

Γ, Δ abbreviates $\Gamma \cup \Delta$.

Γ, ϕ abbreviates $\Gamma \cup \{\phi\}$.

Additionally, we adopt the following metalinguistic terminology.

A piece of metalanguage of the form $\Gamma \vdash \psi$ is called a a **sequent**, read 'Γ proves (*or* deduces, *or* derives) ψ'.

In a sequent $\Gamma \vdash \psi$, Γ is called the **context** and ψ is called the **succedent**.

The members of the context in a sequent are called the **assumptions** or **hypotheses**.

We use Σ (possibly subscripted) as a metavariable over sequents.

Rules are metalanguage conditional assertions that if (a) certain sequent(s) obtain(s), then so does some other sequent.

Rules are written in the form

$$\frac{\Sigma_1 \ \ldots \ \Sigma_n}{\Sigma}$$

where n is 1 or 2.

In a rule, the Σ_i are called the **premisses**, and Σ is called the **conclusion**.

An **axiom** can be thought of as a rule with no premisses; thus it is an (unconditional) assertion that a certain sequent obtains.

Then the recursive definition of the provability relation $\vdash$ is as follows:

TI (Truth Introduction)

$$\frac{}{\vdash T}$$

R (Reflexivity)

$$\frac{}{\phi \vdash \phi}$$

$\wedge$E (Conjunction Elimination 1)

$$\frac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \phi}$$

$\wedge$E' (Conjunction Elimination 2)

$$\frac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \psi}$$

$\wedge$I (Conjunction Introduction)

$$\frac{\Gamma \vdash \phi \qquad \Gamma \vdash \psi}{\Gamma \vdash \phi \wedge \psi}$$

$\rightarrow$E (Implication Elimination)

$$\frac{\Gamma \vdash \phi \qquad \Gamma \vdash \phi \rightarrow \psi}{\Gamma \vdash \psi}$$

$\rightarrow$I (Implication Introduction)

$$\frac{\Gamma, \phi \vdash \psi}{\Gamma \vdash \phi \rightarrow \psi}$$

W (Weakening)

$$\frac{\Gamma \vdash \psi}{\Gamma, \phi \vdash \psi}$$

3 Formal Proofs

We now define a **(formal) proof** of a sequent Σ to be a labelled tree with each node labelled by a sequent, such that the following three condtions hold:

1. The root is labelled by Σ.
2. Each leaf is labelled by an axiom.
3. For each nonleaf x, the label of x is the conclusion of a rule whose premisses are the labels of x's daughters.

Note that, technically, a label is not really a sequent '$\Gamma \vdash \phi$', which is a piece of metalanguage, but rather the ordered pair $\langle \Gamma, \phi \rangle \in \mathsf{Con} \times \Phi$. Nevertheless, as a mnemonic, we notate the labels as '$\Gamma \vdash \phi$', not '$\langle \Gamma, \phi \rangle$'.

It should be intuitively obvious (though we omit the proof) that $\Gamma \vdash \phi$ (the ordered pair $\langle \Gamma, \phi \rangle$ is in the relation $\vdash$) iff there is a (formal) proof of $\langle \Gamma, \phi \rangle$. In that case we call $\Gamma \vdash \phi$ a **(PIPL)-theorem**. Equivalently, we say ϕ is **(PIPL-)provable** (or **deducible**, or **derivable**) **from** Γ, or Γ **proves** (or **deduces**, or **derives**) ϕ (**in PIPL**). Some examples of PIPL-theorems follow, accompanied by formal proofs.

PIPL theorem: $\phi \vdash \phi \wedge \phi$

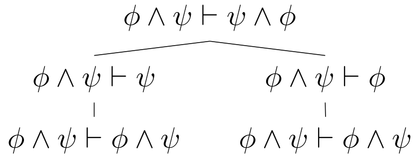
Formal proof:

$$\phi \vdash \phi \wedge \phi$$

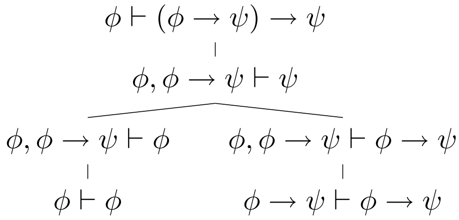
$$\phi \vdash \phi \qquad \phi \vdash \phi$$

PIPL theorem: $\phi \wedge \psi \vdash \psi \wedge \phi$

Formal proof:

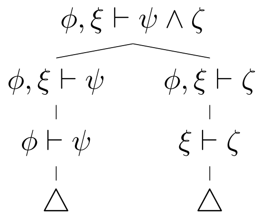


PIPL theorem: $\phi \vdash (\phi \to \psi) \to \psi$
Formal proof:



4 Derived Rules

A **derived rule** is a metalanguage assertion which says that if (a) certain sequent(s) is provable, then so is some other sequent. Derived rules differ from the rules that were used to recursively define the provability relation only by virtue of not being given as part of that definition, but instead having been established as (meta-)theorems. For example, suppose that ϕ, ψ, ξ, and ζ are formulas such that $\phi \vdash \psi$ and $\xi \vdash \zeta$. Then it must also be the case that $\phi, \xi \vdash \psi \wedge \zeta$. The reason is that, *if* we had formal proofs of $\phi \vdash \psi$ and $\xi \vdash \zeta$, indicated below by triangles, *then* the following would be a formal proof of $\phi, \xi \vdash \psi, \zeta$:



In general, the way we establish a derived rule is the same way we establish a formal theorm: by giving a formal proof. The only difference is that in the case of a derived rule, the labels of the leaves in the formal proof can include the premisses of the rule being established, not just axioms.

Once established, a derived rule is notated the same way as one of the original rules, in the present instance:

$$\frac{\phi \vdash \psi \qquad \xi \vdash \zeta}{\phi, \xi \vdash \psi \wedge \zeta}$$

and subsequently be used in formal proofs just as the original rules were.

Once a sequent has been shown to be provable, it can be used to label a leaf of a formal proof just as if it were an axiom. Of course, technically speaking, that node cannot *really* be a leaf of the formal proof, since the formal proof of the sequent in question must be 'hung' under that node as a subtree. But in displaying such a proof, that subtree would normally be represented by a triangle or simply omitted. This is analogous to what we do in informal (metalanguage) proofs: once we prove a lemma, we don't repeat its proof every time we use the lemma to prove another theorem!

5 Semantic Consequence

Recall (Chapter 9) that an HPS interpetation is an HPS $\langle P, \sqsubseteq, \sqcap, \rightarrow, \top \rangle$ together with a function $\mathsf{sem} : \Phi \rightarrow P$ such that:

$$\mathsf{sem}(\phi \wedge \psi) = \mathsf{sem}(\phi) \sqcap \mathsf{sem}(\psi)$$

$$\mathsf{sem}(\phi \rightarrow \psi) = \mathsf{sem}(\phi) \rightarrow \mathsf{sem}(\psi)$$

$$\mathsf{sem}(T) = \top.$$

We now define another relation (besides the provability relation $\vdash$) between contexts and formulas, called the **semantic consequence** relation, written $\models$. Unlike $\vdash$, which was defined in terms of axioms and rules which made reference only to the syntactic forms of formulas, $\models$ is defined semantically, in terms of HPS interpretations, as follows: $\Gamma \models \psi$ iff, for every HPS intepretation, $\mathsf{sem}[\Gamma] \sqsubseteq \mathsf{sem}(\psi)$. This is read '$\psi$ is a semantic consequence of Γ', or 'Γ has ψ as a semantic consequence'. Here, of course, $\sqsubseteq$ is the generalized entailment relation introduced in Chapter 9, section 5.

We use the same abbreviations for contexts in assertions of semantic consequence as we did for assertions of provability (section 2). Some simple facts about semantic conequence, which follow directly from the definitions, are the following;

a. $\models \psi$ iff $\mathsf{sem}(\psi) \equiv \top$ for every HPS interpretation.

b. $\phi \models \psi$ iff $\mathsf{sem}(\phi) \sqsubseteq \mathsf{sem}(\psi)$ for every HPS interpretation.

c. $\phi_1, \ldots, \phi_n \models \psi$ iff $\mathsf{sem}(\phi_1), \ldots, \mathsf{sem}(\phi_n) \sqsubseteq \mathsf{sem}(\psi)$ for every HPS interpretation.

It is a remarkable fact, which we have neither time nor space to prove, that $\vdash$ and $\models$, are the same relation. That is: a set of PIPL formulas proves a formula iff it has that formula as a semantic consequence. The 'only if' and 'if' parts of this assertion constitute, respectively, what are usually called the **soundness** and **completeness** of PIPL deduction with respect to the class of HPS interpretation. Thus the syntax and semantics of PIPL, in a certain precise sense, are 'made for each other'.