Intermediate Logic

Lecture notes

Sandy Berkovski

Bilkent University
Fall 2010

Chapter 3

Sentence calculus: Syntax

We assume that the notion of axiomatic system is understood from §2.3.1. Let us begin by considering the standard axiomatic system for sentence calculus. This system is going to be expressed in a particular language. We shall now describe that language.

Formal languages (or calculi) we shall study have a certain affinity with physical theories. Like those theories, they describe a reality. Their reality is mathematical reasoning or workings of abstract automata. Different language are distinguished by the scope of mathematical reasoning they purport to describe, by their orientation to certain types of mathematical theories, and by their alphabets.

The alphabet of a formal language consists of a collection of symbols, or *letters*. Those symbols form *words*. A word, in general, will be absolutely any collection of letters. For example, 'a' and 'j' are letters, while strings of letters 'mama', 'ghstsr', or 'h6u,8' are words. There are `infinitely` many words to be formed from the given collection of symbols. However, there are only finitely many letters in the alphabets of natural languages. That will not be the case with the formal language we will be interested in: it will have infinitely many letters. why? by repetition

Clearly we are not interested in just any collection of symbols. We need to fix the rules of formation which will segregate between words built correctly, *i.e.* in accordance with those rules, and all other words. The correctly formed words are called *formulae*, or else the *well-formed formulae*, abbreviated as *wff*. Rules of formation are purely syntactic; that is to say, we are not asking a question whether those formulae mean anything. Meaning is not determined at this stage. We understand rules of formation as rules for manipulation with letters. Any string of letters will be well-formed, so far as its letters are arranged in the right order. Intuitively, the words 'mama' and 'ghstsr' composed of the letters of the Latin alphabet will both come out as well-formed, but 'h6u,8' will not.

The main task in investigating formal calculi lies in verifying the truth of certain classes of formulae. We may distinguish between 'syntactic truth' and 'semantic truth'. Semantic truth (or truth *simpliciter*) will be the subject of the next chapter. Syntactic truth is introduced through axioms and rules of inference and is eventually interpreted as 'provability' or 'deducibility'. Refining those notions is our present concern.

3.1 Hilbert system

3.1.1 Formation rules

Let us put forward the rules of formation (or FR) for the sentential language $\mathsf{H_s}$. We shall regard the letters $P_1, P_2, P_3, \ldots$ as standing for declarative sentences, like the sentence 'Socrates walks.' We call them *sentence parameters*. Let the symbols '$\neg$' and '$\supset$' be *sentential connectives*, or *logical constants*. Intuitively we recognise that they stand for negation and implication respectively, but it is important to keep in mind that no such correlation has been made yet. Our alphabet will also include parentheses ')' and '('. The rules are, therefore, as follows: Two groups of rules

FR1. Any of the sentence parameters $P_1, P_2, P_3, \ldots$ is a formula of $\mathsf{H_s}$;

FR2. If A is a formula of $\mathsf{H_s}$, then so is $\neg A$;

FR3. If A and B are formulae of $\mathsf{H_s}$, then so is $(A \supset B)$.

To repeat, there is an essential difference between sentence parameters P_i and formulae $A, B, \ldots$. Whereas sentence parameters stand for simple (atomic) declarative sentences, formulae are arbitrary expressions built up from atomic sentences in accordance with rules of formation.

Let us illustrate the application of formation rules with an example. We want to determine whether a formula $\neg(\neg P \supset \neg(Q \supset P))$ is well-formed. We can write down the following formation sequence:

$$
\begin{array}{ll}
Q & \text{by FR1} \\
P & \text{by FR1} \\
(Q \supset P) & \text{by FR3} \\
\neg(Q \supset P) & \text{by FR2} \\
\neg P & \text{by FR2} \\
(\neg P \supset \neg(Q \supset P)) & \text{by FR3} \\
\neg(\neg P \supset \neg(Q \supset P)) & \text{by FR2.}
\end{array}
$$

Alternatively, utilising our notion of a dyadic tree, we can build a dyadic formation tree for the same formula:

$$
\begin{array}{ccc}
 & \neg(\neg P \supset \neg(Q \supset P)) & \\
 & (\neg P \supset \neg(Q \supset P)) & \\
\neg P & & \neg(Q \supset P) \\
P & & (Q \supset P) \\
 & Q & \qquad P
\end{array}
$$

As an exercise, reflect on the rules for building such a tree. Notice finally that according to our formation rules, the formula $(A \supset B) \supset A$ is not well-formed: it lacks the outward parentheses. In practice, however, we shall be lax and omit the outward parentheses.

3.1.2 Remark on quasi-quotes

Recall the distinction between language and metalanguage. Our sentential parameters P_i belong to the metalanguage. They are constants taking the value of fixed sentential atomic expressions of the object-language $\mathsf{H_s}$. Similarly, the letters $A, B, \ldots$ are meta-linguistic variables. Their value are different formulae of the object-language $\mathsf{H_s}$.

What about sentential connectives? They are plainly part of the object-language, for much the same reason as the expressions 'Not $\cdots$' and 'If $\cdots$, then $\cdots$' are part of English. But then the construction $A \supset B$, legitimised by our formation rules, contains parts of the object-language and the metalanguage all at once. So it is not clear whether it itself belongs to the object-language or the metalanguage. To deal with this mixture, Quine introduced his famous quasi-quotes. The notation of the construction $\ulcorner A \supset B \urcorner$ explicitly indicates that sentential connectives (and parentheses) are mentioned, while meta-linguistic constants and variables are used. Therefore, according to Quine, if we wish to use sentential connectives, it is proper to write as follows:

If A is a formula of $\mathsf{H_s}$, then so is the formula $\ulcorner \neg A \urcorner$.

The use of quasi-quotes is still popular. Unfortunately, it makes sufficiently complicated expressions utterly unreadable. We will simplify our notation a great deal if we adopt the following policy due to Alonzo Church. We shall treat letters like '$\supset$' or ')' as being part of the metalanguage, as names of themselves. Thus, in the meta-linguistic construction '$A \supset A$ is well-formed' the symbol '$\supset$' is mentioned and used at the same time. In this way we avoid the necessity of employing quasi-quotes.

3.1.3 The axioms

Let us proceed with formulating the axioms. The system we will be discussing was first proposed by Lukasiewicz and investigated by Hilbert and others. There are only three axioms:

A1. $A \supset (B \supset A)$

A2. $(A \supset (B \supset C)) \supset (A \supset B \supset (A \supset C))$

A3. $\neg A \supset \neg B \supset (B \supset A)$

However, all those expressions in A1-A3 should be regarded as *schemata*. That is to say, meta-variables A, B, C can take infinitely many object-linguistic values. Thus, the formulae:

$$P_1 \supset ((P_{99} \supset \neg P_{47}) \supset P_1)$$
$$\neg\neg(P_{12} \supset P_{20}) \supset \neg P_{12} \supset (P_{12} \supset \neg(P_{12} \supset P_{20}))$$

will be instances of our axiom-schemata. And there are infinitely more such instances. The system $\mathsf{H_s}$ has one rule of inference, the *modus ponens*:

$$\frac{A \qquad A \supset B}{B}\ .$$

Having understood the axioms and the rule of inference, we can now define the notion of a proof:

Definition 3.1. A *proof* in $\mathsf{H_s}$ is a sequence $\langle A_1, \ldots, A_n\rangle$ of formulae of $\mathsf{H_s}$ such that for every $i \leq n$, either A_i is an axiom of $\mathsf{H_s}$, or there are numbers $j, k < i$ such that A_j is the formula $A_k \supset A_i$. Such a sequence is also said to be the proof of A_n. Provability

What this definition effectively says is that each entry in the proof sequence is either an axiom, or else follows from previous entries by *modus ponens*. We are ready to give a formal definition of another familiar concept.

Definition 3.2. A formula A is a *theorem* of $\mathsf{H_s}$ if there is a proof of A in $\mathsf{H_s}$.

Statements which qualify as theorems are also said to be *provable*. To indicate that A is a theorem we write '$\vdash A$'.

Example 3.3. A widespread device in Northern European poetry was *kenning*. That was an expression which could replace a single word. Their instance include:

Evil-doer = dragon
Dwelling-place = residence
Mail-shirt = armour
Folk-right = possession
Stone-cliff = wall

and so forth. Simple kennings are those no part of which is a kenning. We can *derive* new kennings from a given kenning by replacing one or more words in it by their kennings. This fixes our rule of inference. Simple kennings serve as axioms. Here is a derivation of a complex kenning:

Warrior
Sword-hurler
Battle-fire-hurler
Spear-storm-fire-hurler
Shield-sorceress-storm-fire-hurler
Ship-moon-sorceress-storm-fire-hurler
Shipyard-horse-moon-sorceress-storm-fire-hurler

Now, it is natural to extend our notion of proof to include arbitrary hypotheses on a par with axioms. Such *hypothetical* reasoning constitutes an important part of any theoretical activity: just notice the frequent occurrence of the locutions 'let's assume', 'suppose', and so forth. We easily revise the definition of proof as follows:

Definition 3.4. Let Γ be a set of formulae of $\mathsf{H_s}$. A *deduction* of A_n from Γ in $\mathsf{H_s}$ is a sequence $\langle A_1, \ldots, A_n\rangle$ of formulae of $\mathsf{H_s}$ such that for every $i \leq n$, either A_i is an axiom of $\mathsf{H_s}$, or $A_i \in \Gamma$, or there are numbers $j, k < i$ such that A_j is the formula $A_k \supset A_i$. Deducibility

Example 3.5. A is deducible from $\{\neg A \supset \neg B, B\}$. This fact is symbolised as $\{\neg A \supset \neg B, B\} \vdash A$.

Our very notation suggests that provability and deducibility are related. There is a set of sentences before the turnstile in the case of deducibility, and there is empty space in the case of provability. It seems as though provability is a special case of deducibility. We shall now confirm our notational insight in a rigourous way.

Proposition 3.6. *A is provable if and only if A is deducible from the empty set of assumptions. Formally: $\vdash A$ if and only if $\varnothing \vdash A$.*

Proof. If A is provable, then there exists a proof of A. Let it be an array $\langle B_1, \ldots, B_n \rangle$. Since it is a proof, no hypotheses are found among $B_1, \ldots, B_n$. Thus, by the definition of deduction, A is deduced from the empty set. Conversely, given that $\varnothing \vdash A$, there is a deduction $\langle B_1, \ldots, B_n \rangle$ of A from $\varnothing$. Since the set of hypotheses is empty, there is a proof of A. □

We may also formally articulate an important fact about proofs: namely, that they are finite. This carries over to deducibility in the following claim.

Proposition 3.7. *$\Gamma \vdash A$ iff for some finite set $\Delta \subseteq \Gamma$, $\Delta \vdash A$.*

Proof. From left to right: if $\Gamma \vdash A$, then there is a deduction sequence $\langle B_1, \ldots, B_n \rangle$ of A from Γ. Clearly the sequence is finite. We may then let $\Delta = \Gamma \cap \{B_1, \ldots, B_n\}$. Then Δ is a `finite subset` of Γ. On the other hand, Δ contains exactly the elements of Γ used in the deduction of A. Thus $\langle B_1, \ldots, B_n \rangle$ is a deduction of A from Δ, and therefore, $\Delta \vdash A$. why? ...

From right to left: suppose A is deducible from a finite set $\Delta \subset \Gamma$. That is, there is a corresponding deduction sequence $\langle B_1, \ldots, B_n \rangle$. Clearly it is also a deduction sequence for $\Gamma \vdash A$. Hence $\Gamma \vdash A$. □ why? ...

Finally, let us define for future reference the rest of the familiar propositional logical constants. It is best to treat them as abbreviations of the already introduced expressions. We obtain the following:

Definition 3.8. '$A \vee B$' $\Longleftrightarrow$ '$A \supset B \supset B$'
'$A \wedge B$' $\Longleftrightarrow$ '$\neg(A \supset \neg B)$'
'$A \leftrightarrow B$' $\Longleftrightarrow$ '$(A \supset B) \wedge (A \supset B)$'.

3.2 The deduction theorem

If we are looking to clarify what statements are provable (or deducible from assumptions) in $\mathsf{H_s}$, then the answer may be tricky. Very often there is no straightforward axiomatic proof. For instance, let us show that $\vdash \neg P_1 \supset (P_1 \supset P_2)$:

$\neg P_2 \supset \neg P_1 \supset (P_1 \supset P_2)$	AS3	(1)
$(\neg P_2 \supset P_1 \supset (P_1 \supset P_2)) \supset (\neg P_1 \supset (\neg P_2 \supset \neg P_1 \supset (P_1 \supset P_2)))$	AS1	(2)
$\neg P_1 \supset (\neg P_2 \supset \neg P_1 \supset (P_1 \supset P_2))$	1, 2, MP	(3)
$(\neg P_1 \supset (\neg P_2 \supset \neg P_1 \supset (P_1 \supset P_2))) \supset ((\neg P_1 \supset (\neg P_2 \supset \neg P_1)) \supset (\neg P_1 \supset (P_1 \supset P_2)))$	AS2	(4)
$(\neg P_1 \supset (\neg P_2 \supset \neg P_1)) \supset (\neg P_1 \supset (P_1 \supset P_2))$	3, 4, MP	(5)
$\neg P_1 \supset (\neg P_2 \supset \neg P_1)$	AS1	(6)
$\neg P_1 \supset (P_1 \supset P_2)$	5, 6, MP	(7)

You are welcome to work through this proof. What is clear is that it is neither easily readable, nor intuitive. The situation worsens in more complicated cases. We shall now prove a useful theorem which is called to facilitate axiomatic proofs. Before that, we quickly prove the following lemma:

Proposition 3.9. *$\vdash A \supset A$.*

Proof.

$$(A \supset ((A \supset A) \supset A)) \supset ((A \supset (A \supset A)) \supset (A \supset A))$$
$$A \supset ((A \supset A) \supset A)$$
$$(A \supset (A \supset A)) \supset (A \supset A)$$
$$A \supset (A \supset A)$$
$$A \supset A.$$

Explanations are left as an exercise. □

Now to our theorem:

Proposition 3.10 (Deduction theorem, Herbrand)**.** *If* $\Gamma \cup \{A\} \vdash B$*, then* $\Gamma \vdash A \supset B$*.*

Proof. We must show that if there is a deduction sequence for $\Gamma \cup \{A\} \vdash B$, then there is a deduction sequence for $\Gamma \vdash A \supset B$.

Let $\langle C_1, \ldots, C_n \rangle$ be a deduction of B from $\Gamma \cup \{A\}$. We start by replacing each C_i with $A \supset C_i$. The resulting sequence would no longer be a deduction sequence. If, however, we manage to transform it into a deduction, we are done, since C_n is nothing but B. We have now to supply some missing steps to transform it into a deduction sequence.

We know that each C_i is either an axiom, or is a member of $\Gamma \cup \{A\}$, or else follows by *modus ponens* from some other C_j and C_k. We deal with these three cases separately:

1. If C_i is an axiom, then we insert the following steps before $A \supset C_i$:

$$\begin{aligned} & C_i \\ & C_i \supset (A \supset C_i). \end{aligned}$$

2. If $C_i \in \Gamma \cup \{A\}$, there are two cases to consider:

   (a) If $C_i \in \Gamma$, then we insert the following steps before $A \supset C_i$:

$$\begin{aligned} & C_i \\ & C_i \supset (A \supset C_i). \end{aligned}$$

   (b) If C_i is A then we insert this (compare the procedure in proving Proposition 3.9):

$$\begin{aligned} & (A \supset ((A \supset A) \supset A)) \supset ((A \supset (A \supset A)) \supset (A \supset A)) \\ & A \supset ((A \supset A) \supset A) \\ & (A \supset (A \supset A)) \supset (A \supset A) \\ & A \supset (A \supset A). \end{aligned}$$

3. If C_j is $C_k \supset C_i$, then before $A \supset C_i$ we insert:

$$\begin{aligned} & (A \supset (C_k \supset C_i)) \supset (A \supset C_k \supset (A \supset C_i)) \\ & A \supset C_k \supset (A \supset C_i). \end{aligned}$$

After we have completed the procedure, we obtain a new sequence $\langle D_1, \ldots, D_n \rangle$. We now need to show that it is a deduction sequence for $\Gamma \vdash A \supset B$. It is convenient to distinguish between its members that were inserted in the course of our procedure and the members that were not. Suppose D_l is an entry which was inserted in the course of our procedure. Then it must be either an axiom of $\mathsf{H_s}$, or a member of Γ, or else follows from the previous entries of the sequence by *modus ponens*. why? . . .

Suppose now that D_l was not inserted—that is, D_l is $A \supset C_i$. In cases 1 and 2a D_l is $A \supset C_i$, in the case 2b D_l is $A \supset A$. In all these cases it will follow from the two preceding steps by *modus ponens*. In case 3 we have a similar situation; but here we have to notice that D_l is a consequence of $A \supset C_k$ and $A \supset C_k \supset (A \supset C_i)$ both of which occur previously in the sequence.

Finally, we notice that, since C_n is B and D_m is $A \supset C_n$, D_m is the desired formula $A \supset B$. Therefore, the sequence $\langle D_1, \ldots, D_m \rangle$ is indeed the deduction sequence for $\Gamma \vdash A \supset B$. □

3.3 Consistency

We can now give an account of an important notion of consistency. We stress that this notion is, properly speaking, a syntactic one. Generally, it applies to sets of sentences. It is easier to start with understanding the opposite notion of inconsistency. If our common usage is any guide, then a typical instance would be the set $\{P, \neg P\}$. It exhibits a contradiction. But more generally, we wish to say that an inconsistent set would be such a set from which it is possible to derive a contradiction. An example would be a set $\{P \supset Q, P, \neg Q\}$.

Thus a consistent set is such a set from which contradictions are not deducible. What is a contradiction? It is not a property of a set of sentences; rather, it is a sentence. Here we should follow informal usage and understand it as having the form $A \wedge \neg A$. That is, it will have the form $\neg(A \supset \neg\neg A)$. Conceived this way, a contradiction would simply be a negation of a theorem in $\mathsf{H_s}$, namely, the Proposition 3.9 (since $\neg\neg A \vdash A$). Plainly such a formulation would not exhaust the use of the term of inconsistency in our common parlance. The following lemma helps to see things more generally:

Proposition 3.11. $\{\neg A, A\} \vdash B$.

Proof. We first notice that if $\Gamma \cup \{\neg A\} \vdash \neg B$, then $\Gamma \cup \{B\} \vdash A$. The proof is left as an exercise (hint: use the Deduction theorem). On the other hand, $\{\neg A, \neg B\} \vdash \neg A$. Putting these together, we get the result. □ why? ...

Therefore, an inconsistent set of sentences is such that allows to deduce any sentence whatsoever. Naturally, a consistent set of sentences should not have this property. Hence the desired notion of consistency:

Definition 3.12. A set Γ of formulae of $\mathsf{H_s}$ is *consistent* if there is some formula A of $\mathsf{H_s}$ such that not $\Gamma \vdash A$. A set of sentences is *inconsistent* if it is not consistent.

Armed with this definition, let us prove another familiar proposition:

Proposition 3.13. *A set Γ of formulae in $\mathsf{H_s}$ is inconsistent if and only if for some formula B of $\mathsf{H_s}$, $\Gamma \vdash B$ and $\Gamma \vdash \neg B$.*

Proof. From left to right: If Γ is inconsistent, then $\Gamma \vdash B$ for all B, and in particular, P and $\neg P$.

From right to left: Let us write it down with a little more precision:

$\Gamma \vdash A$	Ass.	(1)
$\Gamma \vdash \neg A$	Ass.	(2)
$\{\neg A, A\} \vdash B$	Prop. 3.11	(3)
$\{\neg A\} \cup \{A\} \vdash B$	3, set theory	(4)
$\Gamma \cup \{\neg A\} \vdash B$	1, 4, Exercise 11.4.2.3	(5)
$\Gamma \cup \Gamma \vdash B$	2, 5, Exercise 11.4.2.3	(6)
$\Gamma \vdash B$	6, set theory	(7)

Since B is any formula, Γ is inconsistent by definition. □

Chapter 4

Sentence calculus: Semantics

So far we have ignored the fact that languages are used to say things about something. To repair this fault we have to resort to the semantic approach.

4.1 Valuation, satisfaction, validity

If sentences say something about something, then they must be regarded as true or false, depending on what they say about what. So we need to assign truth-values to them. Those truth-values may be thought as members of the set $\{\mathbf{1}, \mathbf{0}\}$, where $\mathbf{1}$ and $\mathbf{0}$ are primitive elements. We can conduct the procedure in a systematic way. Sentence parameters standing for true sentences will be assigned the value $\mathbf{1}$, and those standing for false sentences will be assigned the value $\mathbf{0}$. We now have to decide about the way to assign truth-values to complex sentences. We must adopt the *principle of compositionality*: the truth-value of a complex sentences must be determined by the truth-values of its constituents. It is clear that $\neg P$ must be true when P is false, and *vice versa*. Thus the truth-value of $\neg A$ presents little difficulty. It is far less clear how to assign truth-values to the formulae of the form $A \supset B$. This in fact is a difficult issue in the philosophy of logic. Here we shall be content with identifying $A \supset B$ with material implication. We may then fix the truth-values for the rest of logical constants by following abbreviations introduced above. Compositionality

Before we do that, there is one small complication to resolve. If we wish to assign truth-values to the formulae of $\mathsf{H_s}$, the first thing to do is to assign truth-values to sentence parameters. So far we have considered a just one set of sentence parameters. But we may face a situation where the assignment of truth-values to certain sentence parameters leaves other sentence parameters truth-valueless. Equally, however, if we have to decide the truth-value of one particular formula, it would be odd to assign truth-values to all the parameters in $\mathsf{H_s}$ just for this purpose. Thus we need to revise slightly the approach taken so far. Instead of one fixed vocabulary for the system $\mathsf{H_s}$, we may consider an arbitrary vocabulary. The system $\mathsf{H_s}$ will then determine the set of well-formed formulae made up of the parameters in that vocabulary. Hence:

Definition 4.1. A *signature* Σ for $\mathsf{H_s}$ is a non-empty set of sentence parameters.

The notions of a formula, proof, deducibility, all carry over, with an exception that sentence parameters are drawn from Σ.

Therefore, whenever we speak about sentence parameters and formulae, we understand them as belonging to a particular signature. The set of well-formed formulae of the signature Σ of $\mathsf{H_s}$ we designate as $\mathbf{F}_\Sigma$. It is understood that everywhere in this chapter we deal with signatures of $\mathsf{H_s}$.

Definition 4.2. A *valuation* V of the signature Σ for $\mathsf{H_s}$ is a mapping assigning to each sentence parameter of Σ one and only one of the values $\mathbf{1}$ and $\mathbf{0}$. Valuation

Here it is also an opportunity to introduce the notions of *contradiction* and *tautology*. We shall interpret the symbols '$\bot$' and '$\top$' as constants standing for two propositions which are assigned $\mathbf{0}$ and $\mathbf{1}$ respectively by every valuation function. Tautologies are also called *valid* formulae. We also need to fix the rules for assigning truth-values to complex formulae. This is done as follows:

Definition 4.3. Let V be a valuation of the signature Σ for $\mathsf{H_s}$. The truth-value of complex formulae of Σ is determined as follows:

1. $V(\neg A) = \mathbf{1}$ iff $V(A) = \mathbf{0}$;

2. $V(A \supset B) = \mathbf{1}$ iff $V(A) = \mathbf{0}$ or $V(B) = \mathbf{1}$;
3. $V(A \wedge B) = \mathbf{1}$ iff $V(A) = \mathbf{1}$ and $V(B) = \mathbf{1}$;
4. $V(A \vee B) = \mathbf{1}$ iff $V(A) = \mathbf{1}$ or $V(B) = \mathbf{1}$;
5. $V(A \leftrightarrow B) = \mathbf{1}$ iff $V(A) = V(B)$.

Remark. Another way to put some of these clauses is as follows:

1. $V(\neg A) = 1 - V(A)$;
2. $V(A \supset B) = \min(V(\neg A), V(B))$;
3. $V(A \wedge B) = \max(V(A), V(B))$;
4. $V(A \vee B) = \min(V(A), V(B))$.

But here we must require prior understanding of arithmetical operations on truth values.

The next theorem exemplifies an important proof technique know as induction on the complexity of a formula. Before that we define an auxiliary notion.

Definition 4.4. Let $A \in \mathbf{F}_\Sigma$. The *degree* of A is determined as follows:

1. The degree of a sentence parameter is 0;
2. If A has the degree i, then $\neg A$ has the degree $i + 1$;
3. If A and B have the degrees i and j respectively, then the degree of $A \supset B$ is $i + j$.

If A has the degree higher than B, then A is said to be more *complex* than B.

Proposition 4.5. *Let V be a valuation of the signature Σ for $\mathsf{H_s}$. Then every formula $A \in \mathbf{F}_\Sigma$ is assigned a unique value by V.*

Induction on the complexity of a formula

Proof. We induce on the complexity of A. If A is a sentence parameter, $V(A)$ returns a unique value by definition (the basic step of induction). As the inductive hypothesis, suppose that all formulae of Σ less complex than A are assigned a unique truth-value by V. If, therefore, A has the form $B \supset C$, then B and C are assigned a unique value by assumption, while the implication is assigned a unique value by the clause 2 in the definition. If A has the form $\neg D$, then D is assigned a unique value by assumption, while the negation is assigned a unique value by the clause 1 in the definition. □

Let us now define two other important notions.

Satisfiability

Definition 4.6. A formula $A \in \mathbf{F}_\Sigma$ is *satisfiable* if there is a valuation V of Σ such that $V(A) = \mathbf{1}$.

Truth sets

Definition 4.7. Let V be a valuation of Σ. The set of all formulae $A_1, \ldots, A_i, \ldots$ of Σ made true by V is called a *truth set*.

Example 4.8. The formula P is satisfiable. The formula $\neg P$ is also satisfiable. The formula $\neg(P \vee \neg P)$ is not satisfiable. (This is easily checked by examining the relevant truth-tables.)

One may ask at this stage whether satisfiability and validity are signature-relative—that is, whether it is possible for a formula to be satisfiable when regarded as belonging to one signature, and not satisfiable when regarded as belonging to a different signature.

Proposition 4.9. *Let Σ and Σ' be two signatures for $\mathsf{H_s}$ such that $\Sigma' \subseteq \Sigma$. Let $A \in \mathbf{F}_\Sigma$. Let V be a valuation on Σ, and let V' be such that:*

$$V(P) = \begin{cases} V'(P) & \text{if } P \in \Sigma' \\ \text{undefined} & \text{if } P \notin \Sigma'. \end{cases}$$

Then $V(A) = V'(A)$.

Proof. We shall induce on the complexity of A. If A is a sentence parameter, then $V(A) = V'(A)$ by assumption. Suppose now that for all formulae B less complex than A, $V(B) = V'(B)$. If A is an implication $C \supset D$, then $V(A)$ and $V'(A)$ are calculated from $V(C)$, $V(D)$, $V'(C)$, and $V'(D)$. Since the induction hypothesis guarantees that $V(C) = V'(C)$ and $V(D) = V'(D)$, $V(A) = V'(A)$. Similarly for negation. □

Hence:

Proposition 4.10. *Let Σ and Σ' be two signatures for $\mathsf{H_s}$, and let A be a formula of both Σ and Σ'. Then there is a valuation of Σ that satisfies A iff there is a valuation of Σ' that satisfies A.*

Proof. Consider the set $\tilde{\Sigma} = \Sigma \cap \Sigma'$. Then A is a formula of $\tilde{\Sigma}$. Thus, by Proposition 4.9, there is a valuation V of Σ which satisfies A iff there is a valuation of $\tilde{\Sigma}$ which satisfies A. Analogously, there is a valuation V of Σ' which satisfies A iff there is a valuation of $\tilde{\Sigma}$ which satisfies A. Putting this together, there is a valuation V of Σ which satisfies A iff there is a valuation V of Σ' which satisfies A. □

Proposition 4.11. *Let Σ and Σ' be two signatures for $\mathsf{H_s}$, and let A be a formula of both Σ and Σ'. Then every valuation V of Σ satisfies A just in case every valuation V' of Σ' satisfies A.*

Proof. Exercise. □

We are now ready to introduce two novel notions which will later turn out to be semantic analogues of consistency and entailment.

Definition 4.12. Let Γ be a set of formulae of the signature Σ of $\mathsf{H_s}$, V a valuation there. Then V *simultaneously satisfies* Γ if it satisfies every formula of it. Simultaneous satisfiability

Example 4.13. Let $\Gamma = \{P \supset Q, P \supset P, \neg Q\}$. Let $\Sigma = \{P, Q\}$. Let $V_1(P) = \mathbf{1}$, $V_1(Q) = \mathbf{0}$, $V_2(P) = \mathbf{0}$, $V_2(Q) = \mathbf{0}$. Then V_2 simultaneously satisfies Γ, and V_1 does not. (Again, we easily verify this by consulting a relevant truth-table.)

It is clear that simultaneous satisfiability of the members of Γ should guarantee satisfiability of every member of Γ, but not *vice versa*.

Definition 4.14. Let Σ be a signature for $\mathsf{H_s}$, and let $A \in \mathbf{F}_\Sigma$. Then Γ *semantically entails* A, if every valuation of Σ which simultaneously satisfies Γ also satisfies A.

This fact of semantic entailment we designate as '$\Gamma \vDash A$'. In case $\Gamma = \varnothing$ and $\Gamma \vDash A$ we write $\vDash A$. It is easy to see that $\vDash A$ if and only if A is a tautology.

4.2 Normal forms

We shall now establish an interesting fact about a familiar device—the truth-tables. Every formula 'expresses' a truth-table—that is for every formula, we can construct a corresponding truth-table. This should become clear by the way of constructing a truth-table: every row of it displays truth-values taken by sentence parameters. But the opposite is not so clear: what is the guarantee that any truth-table, however randomly arranged, will be expressed by a formula of our sentence calculus? In fact, there is such a guarantee.

Example 4.15. Consider the following truth-table:

P	Q	R	A
1	**1**	**1**	**0**
1	**1**	**0**	**0**
1	**0**	**1**	**1**
1	**0**	**0**	**0**
0	**1**	**1**	**1**
0	**1**	**0**	**0**
0	**0**	**1**	**1**
0	**0**	**0**	**0**

Our question is whether there is a formula A expressing this truth-table. Each row of the truth-table displaying the values of sentence-parameters 'describes', or 'represents', a certain situation. For instance, the second row represents a situation where P and Q are true, and R is false. Thus, looking at the rows of this truth-table with the value **1** in the last column, we may say that they will represent the situations where the desired complex formula A is true. That is, A is true when either of these situations obtains. Accordingly, A is represented as:

$$((P \wedge \neg Q \wedge R) \vee (\neg P \wedge Q \wedge R)) \vee (\neg P \wedge \neg Q \wedge R).$$

A similar procedure can be carried out for conjunction. Here we look at the rows where A is false, and then use De Morgan's laws. Thus the conjunctive form of A:

$$((((\neg(P \wedge Q \wedge R) \wedge \neg(P \wedge Q \wedge \neg R)) \wedge \neg(P \wedge \neg Q \wedge \neg R)) \wedge \neg(\neg P \wedge Q \wedge \neg R)) \wedge \neg(\neg P \wedge \neg Q \wedge \neg R)),$$

whence:

$$(((((\neg P \vee \neg Q \vee \neg R) \wedge (\neg P \vee \neg Q \vee R)) \wedge (\neg P \vee Q \vee R)) \wedge (P \vee \neg Q \vee R)) \wedge (P \vee Q \vee R)).$$

Definition 4.16. Let A and B be the formulae of Σ. A and B are said to be *logically equivalent* if $\{A\} \vDash B$ and $\{B\} \vDash A$.

The fact of logical equivalence we designate as '$A \simeq B$'. From the definition of logical equivalence it should follow that $A \simeq B$ just in case A and B have the same truth-tables.

Definition 4.17. A formula is said to be in *disjunctive normal form* if it is of the form $A_1 \vee \cdots \vee A_n$, where each formula A_i is of the form $B_1 \wedge \cdots \wedge B_m$, and each B_j is either a sentence parameter, or the negation of a sentence parameter. DNF

Definition 4.18. A formula is said to be in *conjunctive normal form* if it is of the form $A_1 \wedge \cdots \wedge A_n$, where each formula A_i is of the form $B_1 \vee \cdots \vee B_m$, and each B_j is either a sentence parameter, or the negation of a sentence parameter. CNF

Proposition 4.19. *Let $A \in \mathbf{F}_\Sigma$ such that A is not a contradiction. Then there is a formula $B \in \mathbf{F}_\Sigma$ such that B is in disjunctive normal form, and $A \simeq B$.*

Proposition 4.20. *Let $A \in \mathbf{F}_\Sigma$ such that A is not a contradiction. Then there is a formula $B \in \mathbf{F}_\Sigma$ such that B is in conjunctive normal form, and $A \simeq B$.*

The upshot of this discussion is that a language using only negation and disjunction, or only negation and conjunction, as its logical connectives, is *expressively complete* with respect to truth-tables. In other words, if we build well-formed formulae using only the connectives just mentioned, then all truth-tables of two or more columns will be found among the truth-tables of the resulting wffs. Expressive completeness

Another way of putting this fact is to say that the sets $\{\neg, \vee\}$ and $\{\neg, \wedge\}$ are both expressively complete. However, not just any set of connectives is expressively complete.

Proposition 4.21. *The set $\{\supset\}$ is expressively incomplete.*

Proof. We have to show that no combination of wffs in a language L involving only implication as its connective will match every possible truth-table. Such a language will contain as its wffs sentence variables and complex formulae generated by the following rule: if A and B are formulae, then so is $(A \supset B)$. To prove our claim, we must find a truth-table not expressible by any of the formulae of that language. Intuitively, recalling the truth-table for implication, our task will be reduced to finding a truth-table which has the value $\mathbf{0}$ in its last column, but where all sentence parameters are assigned the value $\mathbf{1}$.

To do this rigorously, we must first show that every formula A of the language L is assigned $\mathbf{1}$ when all of its sentence parameters are assigned $\mathbf{1}$. We can induce on the complexity of A. Let V be such a valuation of the signature Σ that it assigns $\mathbf{1}$ to every sentence parameter of Σ. The basis step of induction consists in letting A be a sentence parameter. Then $V(A) = \mathbf{1}$ by assumption. Let A be $B \supset C$ and assume as the inductive hypothesis that for every formula D less complex than A, $V(D) = \mathbf{1}$. Therefore, $V(B) = V(C) = \mathbf{1}$. And now, from the valuation rule for implication it follows that $V(A) = \mathbf{1}$.

Given this result, all we have to do is to find a truth-table which contains a row with the value $\mathbf{1}$ for all sentence parameters and the value $\mathbf{0}$ for the complex formula. The truth-table for negation does the trick. This means that negation is not expressible, or 'definable', in terms of implication. $\square$

We conclude by listing two further definitions.

Definition 4.22. A set X of logical connectives is called *independent* if no connective $x \in X$ can be expressed by the set $X - \{x\}$ of logical connectives.

If X is itself complete, then we can re-write the definition slightly:

Definition 4.23. A complete set X of logical connectives is called *independent* if no proper subset of it is complete.

4.3 The method of semantic tableaux

Tableaux are a tool for proving formulae that utilises the reasoning of *reductio ad absurdum*. To prove the formula A we assume that $\neg A$ and try to derive a contradiction. Given the soundness and completeness of this method, we can also regard tableaux as testing validity and satisfiability. We shall now describe the method rigourously.

Definition 4.24. Let $A \in \mathbf{F}_\Sigma$. A *signed* formula is an expression having the form of $\mathbf{T}A$ or $\mathbf{F}A$. Under any interpretation, the truth value of $\mathbf{T}A$ is the same as that of A, and the truth value of $\mathbf{F}A$ is the same as that of $\neg A$. 'Unsigned formula' is synonymous with 'formula'. A *conjugate* pair of formulae is a pair $\langle \mathbf{T}A, \mathbf{F}A \rangle$.

Definition 4.25. A *signed tableau* is an ordered dyadic tree whose points are occurrences of signed formulae.

We can now define a tableau for a formula A.

Definition 4.26. Let $\mathscr{T}_1$ and $\mathscr{T}_2$ be two signed tableaux. Then $\mathscr{T}_2$ is an *immediate extension* of $\mathscr{T}_1$ if $\mathscr{T}_2$ is obtained from $\mathscr{T}_1$ by applying one of the tableau processing rules in Table 4.3 to a finite path of $\mathscr{T}_1$.

$$
\begin{array}{c}\vdots\\ \mathbf{T}\neg A\\ \vdots\\ \\ \mathbf{F}A\end{array}
\qquad
\begin{array}{c}\vdots\\ \mathbf{F}\neg A\\ \vdots\\ \\ \mathbf{T}A\end{array}
\qquad
\begin{array}{cc}\multicolumn{2}{c}{\vdots}\\ \multicolumn{2}{c}{\mathbf{T}A \supset B}\\ \multicolumn{2}{c}{\vdots}\\ \\ \mathbf{F}A & \mathbf{T}B\end{array}
\qquad
\begin{array}{c}\vdots\\ \mathbf{F}A \supset B\\ \vdots\\ \\ \mathbf{T}A\\ \mathbf{F}B\end{array}
$$

$$
\begin{array}{c}\vdots\\ \mathbf{T}A \wedge B\\ \vdots\\ \\ \mathbf{T}A\\ \mathbf{T}B\end{array}
\qquad
\begin{array}{cc}\multicolumn{2}{c}{\vdots}\\ \multicolumn{2}{c}{\mathbf{F}A \wedge B}\\ \multicolumn{2}{c}{\vdots}\\ \\ \mathbf{F}A & \mathbf{F}B\end{array}
\qquad
\begin{array}{cc}\multicolumn{2}{c}{\vdots}\\ \multicolumn{2}{c}{\mathbf{T}A \vee B}\\ \multicolumn{2}{c}{\vdots}\\ \\ \mathbf{T}A & \mathbf{T}B\end{array}
\qquad
\begin{array}{c}\vdots\\ \mathbf{F}A \wedge B\\ \vdots\\ \\ \mathbf{F}A\\ \mathbf{F}B\end{array}
$$

$$
\begin{array}{cc}\multicolumn{2}{c}{\vdots}\\ \multicolumn{2}{c}{\mathbf{T}A \leftrightarrow B}\\ \multicolumn{2}{c}{\vdots}\\ \\ \mathbf{T}A & \mathbf{F}A\\ \mathbf{T}B & \mathbf{F}B\end{array}
\qquad
\begin{array}{cc}\multicolumn{2}{c}{\vdots}\\ \multicolumn{2}{c}{\mathbf{F}A \leftrightarrow B}\\ \multicolumn{2}{c}{\vdots}\\ \\ \mathbf{T}A & \mathbf{F}A\\ \mathbf{F}B & \mathbf{T}B\end{array}
$$

Table 4.1: Signed tableau processing rules

Definition 4.27. A tree $\mathscr{T}$ is a *tableau for A* if and only if there is a sequence $\langle \mathscr{T}_1, \ldots, \mathscr{T}_n \rangle$ such that $\mathscr{T}_1$ is a one-point tree with the origin A, $\mathscr{T}_n = \mathscr{T}$, and for each $i < n$, $\mathscr{T}_{i+1}$ is an immediate extension of $\mathscr{T}_i$.

We can also give a more general formulation of this idea:

Definition 4.28. A tree $\mathscr{T}$ is a *tableau starting with* $A_1, \ldots, A_k$ if and only if there is a sequence $\langle \mathscr{T}_1, \ldots, \mathscr{T}_k, \ldots, Tr_n \rangle$, where $1 \leq k \leq n$, such that, for all $j \leq k$, $\mathscr{T}_j$ is a one-point tree with the origin A_j, $\mathscr{T}_n = \mathscr{T}$, and for each $i < n$, $\mathscr{T}_{i+1}$ is an immediate extension of $\mathscr{T}_i$.

Definition 4.29. A branch θ of a tableau $\mathscr{T}$ is *closed* if it contains a conjugate pair of formulae. We shall indicate that θ is closed by inserting a cross $\times$ into it. The tableau $\mathscr{T}$ is closed if all of its branches are closed.

We can finally specify the application of the tableau method. It consists in the following:

1. To test a formula A for validity, we form a signed tableau for $\mathbf{F}A$. If the tableau closes, then A is logically valid.

2. To test whether B is semantically entailed by $\{A_1, \ldots, A_k\}$, we form a signed tableau starting with $\mathbf{T}A_1, \ldots, \mathbf{T}A_k, \mathbf{F}B$. If the tableau closes, then B is semantically entailed by $\{A_1, \ldots, A_k\}$.

3. To test whether the set $\{A_1, \ldots, A_k\}$ is simultaneously satisfiable, we form a signed tableau starting with $\mathbf{T}A_1, \ldots, \mathbf{T}A_k$. If the tableau closes, then $\{A_1, \ldots, A_k\}$ is *not* simultaneously satisfiable. If the tableau does not close off, then $\{A_1, \ldots, A_k\}$ is satisfiable. Moreover, by looking at any open path we can at once identify a valuation simultaneously satisfying $\{A_1, \ldots, A_k\}$.

These applications, however, are all semantic, and as such, they must be justified. The tableau technique is a *proof* technique operating at the syntactic level. To apply it at a semantic level we must show the correlation between syntactic entailment, as determined by tableau rules, and semantic entailment.

Example 4.30. Let us find a proof for $(P \supset Q) \vee (P \supset R) \vdash P \supset (Q \vee R)$:

$$
\begin{array}{cccc}
\multicolumn{4}{c}{\mathbf{T}(P \supset Q) \vee (P \supset R)} \\
\multicolumn{4}{c}{\mathbf{F}P \supset (Q \vee R)} \\
\multicolumn{4}{c}{\mathbf{T}P} \\
\multicolumn{4}{c}{\mathbf{F}Q \vee R} \\
\multicolumn{4}{c}{\mathbf{F}Q} \\
\multicolumn{4}{c}{\mathbf{F}R} \\
\multicolumn{2}{c}{\mathbf{T}P \supset Q} & \multicolumn{2}{c}{\mathbf{T}P \supset R} \\
\mathbf{F}P & \mathbf{T}Q & \mathbf{F}P & \mathbf{T}R \\
\times & \times & \times & \times
\end{array}
$$

4.4 Soundness and completeness

We wish now to relate syntactic and semantic notions. We may start by comparing tautologies with theorems. Tautologies are true in every situation, whatever the circumstances. Situations here are identified by assigning truth-values to sentence parameters. Any such situation (*viz.* valuation) is legitimate. Therefore, many statements judged to be necessarily true scientifically or mathematically, such as '$7 + 5 = 12$', or 'Water is H_2O', will not come out as tautologies. Indeed, there is no other way to symbolise them than simply as P. On the other hand, there are some formulae, such as $P \supset P$, which will be tautologous apparently in virtue of their *logical* composition, the way their ingredients are related by logical connectives.

Now, a good axiomatic logical theory, *i.e.* a theory characterised by its deductive machinery, must not have among its theorems sentences which could be false. For otherwise, among the sentences that could be false we may also find sentences which are actually false. A theory which proves false claims will not be of much use. Such a theory will not be *sound*. *Vice versa*, it is also desirable that the theory would also prove all of the tautologies. If it fails, it will be *incomplete*.

We wish to align theorems with tautologies, provability with truth in every situation. More generally, we shall also align deducibility ('syntactic entailment') with semantic entailment.

Proposition 4.31. *If* $\vdash A$*, then* $\vDash A$*.*

Proof. By inspecting relevant truth-tables, it is easy to verify that every axiom of $\mathsf{H_s}$ is a tautology. On the other hand, from the definition of satisfiability and the truth-table for material implication it follows that if $\vDash A$ and $\vDash A \supset B$, then $\vDash B$. Thus every theorem of $\mathsf{H_s}$, obtained by *modus ponens*, will be a tautology. □

In order to prove a more general soundness result let us record two more properties of semantic entailment.

Proposition 4.32. *If* $\Gamma \vDash A \supset B$*, then* $\Gamma \cup \{A\} \vDash B$*.*

Proof. Suppose that $\Gamma \vDash A \supset B$. And suppose, for *reductio*, that $\Gamma \cup \{A\} \nvDash B$. Then there is a valuation V such that V simultaneously satisfies $\Gamma \cup \{A\}$ and $V(B) = \mathbf{0}$. Then $V(A) = \mathbf{1}$. Therefore, $V(A \supset B) = \mathbf{0}$. Since V simultaneously satisfies Γ, it follows that $\Gamma \nvDash A \supset B$, contrary to our assumption. □

Proposition 4.33. *If* $\Gamma \vDash A$*, then* $\Gamma \cup \Delta \vDash A$*.*

Proof. Straightforward from the definition of entailment. □

We are now ready to prove the general result.

Proposition 4.34 (Soundness)**.** *If* $\Gamma \vdash A$*, then* $\Gamma \vDash A$*.*

Proof. Suppose that $\Gamma \vdash A$. By Proposition 3.7 we have a finite set $\Gamma' \subseteq \Gamma$, such that $\Gamma' \vdash A$. Let $\Gamma' = \{B_1, \ldots, B_n\}$. Hence $\{B_1\} \cup \cdots \cup \{B_n\} \vdash A$. Repeatedly applying the Deduction theorem, we get $\vdash B_1 \supset \cdots \supset B_n \supset A$. Then, by Proposition 4.31, $\vDash B_1 \supset \cdots \supset B_n \supset A$. Repeatedly applying Proposition 4.32, we get $\{B_1\} \cup \cdots \cup \{B_n\} \vDash A$, whence $\Gamma' \vDash A$. Therefore, by Proposition 4.33, $\Gamma \vDash A$. □

Practical use of soundness

The results we have achieved so far might not seem too spectacular, but they help in practical matters. The rule of contraposition allows us to read the soundness theorem as saying that if $\Gamma \nvDash A$, then $\Gamma \nvdash A$. Therefore, we can easily show, by examining the relevant truth tables, that some formulae are not provable (if $\Gamma = \varnothing$) and that some formulae are not deducible.

Example 4.35. To show that $\{P \supset (P \wedge Q), Q\} \nvdash P$, one should find a row in the truth table where $V(P \supset (P \wedge Q)) = V(Q) = \mathbf{1}$ and $V(P) = \mathbf{0}$—a purely mechanical procedure given the finite nature of truth tables.

Properties of truth sets

We now move on to a more difficult task of showing the completeness of $\mathsf{H_s}$. We shall achieve this by reflecting on the properties of truth sets introduced earlier. We start by listing some of their immediate properties.

Proposition 4.36. *Let* $A \in \mathbf{F}_\Sigma$*. Let* Γ *be a truth set of* Σ*. Then:*

1. *If* A *is of the form* $\neg B$*, then* $A \in \Gamma$ *just in case* $B \notin \Gamma$*;*
2. *If* A *is of the form* $B \wedge C$*, then* $A \in \Gamma$ *just in case* $B \in \Gamma$ *and* $C \in \Gamma$*;*
3. *If* A *is of the form* $B \vee C$*, then* $A \in \Gamma$ *just in case* $B \in \Gamma$ *or* $C \in \Gamma$*;*
4. *If* A *is of the form* $B \supset C$*, then* $A \in \Gamma$ *just in case* $B \notin \Gamma$ *or* $C \in \Gamma$*.*

Proof. Exercise. □

Proposition 4.37. *Let* Γ *be a truth set of* Σ*. Then* Γ *is consistent.*

Proof. Follows from the definition of consistency and the clause 1 of Proposition 4.36. □

We can now uncover further properties of truth sets.

Proposition 4.38. *Let* $A \in \mathbf{F}_\Sigma$*. Let* Γ *be a truth set of* Σ*. If* $\Gamma \vdash A$*, then* $A \in \Gamma$*.*

Proof. By *reductio ad absurdum*:

Γ is a truth set	Ass.	(1)
$\Gamma \vdash A$	Ass.	(2)
$A \notin \Gamma$	Ass.	(3)
$\neg A \in \Gamma$	(3), clause 1 of Prop. 4.36	(4)
Γ is inconsistent	(4), Prop. 3.13	(5)
Γ is consistent	(1), Prop. 4.37	(6)
Contradiction	(5), (6).	(7)

Thus $A \in \Gamma$. □

Proposition 4.39. *Let $A \in \mathbf{F}_\Sigma$. Let Γ be a truth set of Σ. $\Gamma \vdash A$ if and only if $A \in \Gamma$.*

Proof. Straightforward. □

Proposition 4.40. *Let Γ be a truth set of Σ. Let $\Delta \subseteq \mathbf{F}_\Sigma$ be a consistent set such that $\Gamma \subseteq \Delta$. Then $\Gamma = \Delta$.*

Proof. Suppose, for *reductio*, that $\Gamma \neq \Delta$. Then there is a formula A such that $A \notin \Gamma$ and $A \in \Delta$. But then $\neg A$ is in Γ by the clause 1 of Proposition 4.36. And then Δ is inconsistent, contrary to the assumption. □

Therefore, truth sets are always *maximal consistent sets*: adding any formula to them will render them inconsistent.

Although we have said that every truth set is consistent, the reverse is not true. Let $\Sigma = \{P, Q\}$ and let $\Gamma = \{P, \neg Q\}$. Then Γ is consistent, but is not a truth set: it is not maximally consistent (for instance, it contains neither $P \supset Q$, nor $\neg(P \supset Q)$). But suppose we enlarge Γ so, that we consider its deductive closure—the set of all sentences deducible from Γ. The resulting set `will in fact be a truth set`. However, would it be the case that the deductive closure of any consistent set is a truth set? No: let $\Delta = \{A \mid A \in \mathbf{F}_\Sigma \text{ and } P \vdash A\}$. Then Δ is clearly not a truth set. why? ...

And yet, even in that last case we can easily enlarge Δ by adding $\neg Q$ and then take its deductive closure. And this is true in general: every consistent set is contained in a truth set. Proving it will be a major step towards proving completeness. The method is intuitive: what we should do is to enlarge the original consistent set by adding formulae one by one, and thereby constructing a sequence of consistent sets. Each member of that sequence will be a subset of later members. We then have to show that the resulting infinite union formed out of the members of our sequence contains either A or $\neg A$. But first we have to prove that the union is consistent.

Proposition 4.41. *Let $\Gamma_1, \Gamma_2, \dots$ be a sequence of consistent sets of formulae of Σ such that $\Gamma_i \subseteq \Gamma_j$ for any $i < j$. Let $\Delta = \Gamma_1 \cup \Gamma_2 \cup \cdots$. Then Δ is consistent.*

Proof. Let $\Delta' = \{A_1, \dots, A_n\} \subseteq \Delta$. Then for each A_k there is a set $\Gamma_{f(k)}$ such that $A_k \in \Gamma_{f(k)}$. Let m be largest number of $f(1), \dots, f(n)$. Since $\Gamma_i \subseteq \Gamma_j$ for $i < j$, we have that $\Delta' \subseteq \Gamma_m$. Since Γ_m is consistent, Δ' is consistent, too. And since every finite subset of Δ is a subset of Γ_i for some i, it is consistent. Then Δ is consistent by Exercise **??**.**??**. □

Proposition 4.42 (Lindenbaum)**.** *Every consistent set of formulae of Σ is contained in a truth set.*

Proof. Let $\Gamma \subseteq \mathbf{F}_\Sigma$. Let $A_1, \dots, A_n$ be an ordering of all the formulae of Σ. We define the sequence $\Gamma_1, \Gamma_2, \dots$ according to following rule:

$$\Gamma_{i+1} = \begin{cases} \Gamma_i \cup \{A_i\} & \text{if } \Gamma_i \cup \{A_i\} \text{ is consistent} \\ \Gamma_i \cup \{\neg A_i\} & \text{if } \Gamma_i \cup \{A_i\} \text{ is inconsistent.} \end{cases}$$

To construct the set Γ_{i+1} we have to test A_i for consistency with Γ_i.

We show by induction on i that Γ_i is consistent for every i. $\Gamma_1 = \Gamma$ is consistent by assumption. Suppose Γ_i is consistent. Then, by Exercise **??**.**??** and definition of consistency, either $\Gamma_i \cup \{A_i\}$ is consistent, or $\Gamma_i \cup \{\neg A_i\}$ is consistent. If $\Gamma_i \cup \{A_i\}$ is consistent, then $\Gamma_{i+1} = \Gamma_i \cup \{A_i\}$ by definition, and so Γ_{i+1} is consistent. If $\Gamma_i \cup \{A_i\}$ is inconsistent, then $\Gamma_i \cup \{\neg A_i\}$ is consistent, and so again, $\Gamma_{i+1} = \Gamma_i \cup \{\neg A_i\}$ and Γ_{i+1} is consistent.

Let $\Delta = \Gamma_1 \cup \Gamma_2 \cup \cdots$ and let $B \in \mathbf{F}_\Sigma$. Then $B = A_j$ for some j. Since $B \in \Gamma_{j+1}$ or $\neg B \in \Gamma_{j+1}$, we have that $B \in \Delta$ or $\neg B \in \Delta$. By Proposition 4.41 Δ is consistent, and so Δ is a truth set. □

As a corollary, we have:

Proposition 4.43. *A set of formulae is consistent if and only if it is contained in a truth set.*

On the way to proving completeness, we are now ready first to align consistency with simultaneous satisfiability. Consistency linked to sim. sat.

Proposition 4.44. *Let $\Gamma \subseteq \mathbf{F}_\Sigma$. Then Γ is simultaneously satisfiable if and only if Γ is consistent.*

Proof. From left to right: Suppose, for *reductio*, that Γ is not consistent. Then, by Proposition 3.13, $\Gamma \vdash A$ and $\Gamma \vdash \neg A$. Then, by Proposition 4.34, $\Gamma \vDash A$ and $\Gamma \vDash \neg A$. Then, by definition of semantic entailment, every valuation V which satisfies Γ, must also satisfy A and $\neg A$. That is, $V(A) = V(\neg A)$. But this is impossible by Definition 4.3.

From right to left: Suppose that Γ is consistent. Then, by Proposition 4.42, it is contained in a truth set Δ. Thus, if $B \in \Gamma$, then $B \in \Delta$ for any $B \in \mathbf{F}_\Sigma$. Then there is a valuation V, such that $V(B) = \mathbf{1}$. So V satisfies every formula of Γ; hence Γ is simultaneously satisfiable. □

We can now finally establish completeness. In fact, it is easier to establish completeness along with soundness, and register completeness separately as corollary.

Proposition 4.45. *Let $\Gamma \subseteq \mathbf{F}_\Sigma$. Let A be a formula of Σ. Then $\Gamma \vdash A$ if and only if $\Gamma \vDash A$.*

Proof. By Exercise **??.??**, $\Gamma \vdash A$ iff $\Gamma \cup \{\neg A\}$ is inconsistent. By Proposition 4.44, $\Gamma \cup \{\neg A\}$ is inconsistent iff $\Gamma \cup \{\neg A\}$ is not simultaneously satisfiable. It is easy to see, by inspecting the left-to-right part of the proof of Proposition 4.44, that $\Gamma \cup \{\neg A\}$ is not simultaneously satisfiable iff $\Gamma \vDash A$. Putting all these bi-conditionals together, we have that $\Gamma \vdash A$ iff $\Gamma \vDash A$. $\square$

The promised corollary is this:

Proposition 4.46 (Completeness)**.** *If $\Gamma \vDash A$, then $\Gamma \vdash A$.*

Other important corollaries include:

Proposition 4.47. *$\vdash A$ if and only if $\vDash A$.*

Proposition 4.48. *If $\vDash A$, then $\vdash A$.*

Remark. In our proof of Proposition 4.42 and all other proofs relying on it we assume that there are denumerably many formulae of Σ, so that they can be ordered by natural numbers. If there are non-denumerably many formulae of Σ, another technique is required, one that relies on Zorn's Lemma (or the axiom of choice). It falls far outside the scope of our interest here.

4.5 Compactness

We can now obtain another significant result, that of *compactness*.

Proposition 4.49 (Compactness)**.** *Let $\Gamma \subseteq \mathbf{F}_\Sigma$. Then Γ is simultaneously satisfiable if and only if every finite subset of Γ is simultaneously satisfiable.*

Proof. By applying Proposition 4.44 (twice) and Exercise **??.??**. $\square$

Another form of the compactness theorem is as follows:

Proposition 4.50. *Let $\Gamma \subseteq \mathbf{F}_\Sigma$. Then there is a finite set $\Delta \subseteq \Gamma$ such that if $\Gamma \vDash A$, then $\Delta \vDash A$.*

Proof. Exercise. $\square$

4.6 Gentzen systems

To be omitted. May be introduced at the end.

[Bos97, Men64, Smu68]