Intermediate Logic

Lecture notes

Sandy Berkovski

Bilkent University
Fall 2010

Abstract

The course falls into three parts. In the first part, encompassing Chapters 2-6, we shall streamline and upgrade the treatment of sentence and predicate calculi already familiar from the introductory course. In the second part, which is Chapter **??**, we shall prove the completeness of the predicate calculus utilising methods and techniques acquired in the first part. In the third part, consisting of Chapters 8-9, we shall discuss some elements of the theory of algorithms. We shall then make inroads into *metamathematics* and prove Gödel's famous incompleteness theorems. Finally, in Chapter 10, we shall formulate some important model-theoretic results, but omit most of the proofs.

Disclaimer. These notes do not constitute an attempt of writing a book. Some of the issues discussed here may have been amended, expanded, or further explained in the class.

Chapter 1

Paraphrase

1.1 Paraphrase: connectives

Let us rehearse some topics from the introductory course. We paraphrase declarative sentences that are either true or false into sentences of formal logic. Many ordinary sentences are not declarative, and many declarative sentences are not either true or false.

Example 1.1. 'Would you like some tea?' is not a declarative sentence. 'I am hot' is a declarative sentence, but does not have a truth value. 'Ankara is pretty' is a declarative sentence, but again does not have a truth value. why? ...

In order for a (declarative) sentence to have a truth value it must be supplemented with different contextual parameters. Those may include time, place, or the author of its utterance. Thus 'Ankara is pretty' must be transformed into 'Ankara is pretty in 2010', or perhaps even into 'Ankara is pretty-according-to-UN-standard-of prettiness in 2010'. why? ...

These complications are significant philosophically, but are less so logically. We shall in fact ignore them and treat all grammatically declarative sentences as if they were supplemented with contextual parameters and thus truth-evaluable.

The paraphrase of ordinary language sentences into the sentences of formal logic is to a very large extent arbitrary. The 'correctness' depends on the purposes at hand.

1.2 Paraphrase: quantifiers

Sometimes paraphrase is straightforward.

Example 1.2. Let us paraphrase the sentence 'Everyone is happy' into the language of quantifiers. Let P be the predicate '① is happy'. Then we have:

$$\forall x Px.$$

Let us paraphrase the sentence 'There is someone unhappy'. We have:

$$\exists x \neg Px.$$

But already here we have a choice. Let Q be the predicate '① is unhappy'. Then:

$$\exists x Qx.$$

Which option is better? The better option is the one which reflects negation in the original sentence.

In most cases paraphrase is not quite trivial.

Example 1.3. 'Every philosopher is virtuous.' We immediately have a choice. We can take the domain of objects over which our variables range to be the set of philosophers. Then we have simply: Domain of variables

$$\forall x Vx.$$

There is nothing logically wrong with this decision. But this is not a healthy policy. Normally we want to represent the linguistic structure of the original sentence in our paraphrase. And the original sentence has the predicate '① is a philosopher'. Now we cannot paraphrase this sentence as: Why?

$$\forall x(Px \wedge Vx).$$

We must use implication:

$$\forall x(Px \supset Vx).$$

Example 1.4. 'Every Greek philosopher is virtuous.' If we take the domain to include Greek people, then we get the paraphrase as in the previous example. Naturally we want a different paraphrase. So our domain will include all people:

$$\forall x((Px \wedge Gx) \supset Vx).$$

Let us try something more complicated.

Example 1.5. 'Whenever the Fed cuts the interest rate by more than quarter a percent, the market shares rise and the dollar weakens in relation to all major currencies.' We make the following stipulations:

P : ① cuts the interest rate by ② at ③
Q : ① rises at ②
R : ① weakens in relation to ② at ③
S : ① is bigger than ②
T : ① is a moment of time
U : ① is a major currency
V : ① is a market share
a : 0.25%
b : US Dollar
c : Federal Reserve.

Then we have:

What is our domain?

$$\forall x(Tx \supset \forall y(Sya \supset (Pcyx \supset (\forall z(Vz \supset Qzx) \wedge \forall u(Uu \supset Rbux))))).$$

Now to existential quantification.

Example 1.6. 'Some politicians are virtuous.' The paraphrase is achieved thus:

$$\exists x(Px \wedge Vx).$$

Why aren't we using implication here? The sentence:

$$\exists x(Px \supset Vx)$$

in effect says that someone is either not a politician or is virtuous. This is true even when all politicians are evil.

Example 1.7. 'Nobody in Bilkent invests in Swedish krona.' We have:

$$\neg\exists x(Bx \wedge Ixa).$$

Note that this is equivalent to:

$$\forall x(Bx \supset \neg Ixa).$$

We move to consider cases involving both existential and universal quantification.

Example 1.8. 'Those who have tall girlfriends are tall.' The domain is the set of men. We have:

$$\forall x(\exists y(Gyx \wedge Ty) \supset Tx).$$

Interestingly, that the sentence 'Those who have *only* tall girlfriends are tall' is rendered differently:

$$\forall x(\exists y(Gyx \supset Ty) \supset Tx).$$

Example 1.9. 'Policemen know what criminals are up to.' The domain is the set of people and plans. We make the following stipulations:

Opaque context!

P : ① is a policeman
C : ① is a criminal
U : ① is up to ②
K : ① knows that ② is up to ③.

It is ambiguous what the sentence asserts. First guess:

$$\forall x(Px \supset \forall y \forall z((Cy \wedge Uyz) \supset Kxyz)).$$

But this sentence says that every policeman knows what *every* criminal is up to. Second guess:

$$\forall x(Px \supset \exists y \forall z((Cy \wedge Uyz) \supset Kxyz)).$$

This sentence says that every policeman knows what some criminal is up to. Yet one can take the original sentence to imply that all the criminals are under the control of policemen. Third guess:

$$\forall x(Cx \supset \exists y(Py \wedge \forall z(Uyz \supset Kyxz))).$$

Now this sentence fares better. But why to assume that one policeman must know *everything* the criminal is up to? Fourth guess:

$$\forall x(Cx \supset \forall y(Uyx \supset \exists y(Py \wedge Kzxy))).$$

It seems that this sentence conveys most adequately the meaning of the original English sentence.

Example 1.10. 'All the passengers on that flight lost some of their luggage.' Here it is important to symbolise the relations properly, even though not all of them are reflected in the surface structure of the sentence. We stipulate:

P : ① is a passenger on ②
L : ① is a piece of luggage
B : ① belongs to ②
R : ① lost ②
a : That flight.

Then we get:

$$\forall x(Pxa \supset \exists y(Ly \wedge Byx \wedge Rxy)).$$

1.3 Interpretation

When we interpret a formula of predicate calculus, we generally do the following:

1. Fix the domain.
2. Correlate each sentence parameter with a sentence.
3. Correlate each individual parameter with an individual in the domain.
4. Correlate each predicate parameter with a predicate.

One may notice that we have simply reversed the above procedure of translating sentences of a natural language (English) into the language of predicate calculus.

An uninterpreted formula cannot be assigned a truth value. It is only after we have fixed an interpretation that we can enquire whether a formula is true or false *in* an interpretation.

Example 1.11. Consider the formula $\forall x Px$. Consider the interpretation:

M : the set of natural numbers
P : ① is a number.

Given this interpretation, the formula says that whatever natural number you select, that natural number is a number. Clearly, the formula is true on this interpretation. Consider another interpretation:

M : the set of Bilkent students
P : ① is tall.

Given this interpretation, the formula says that whatever Bilkent student you select, that student is tall. Clearly, the formula is false on this interpretation.

Sometimes we wish to know whether a formula is valid—that is, whether it is true under *every* interpretation. We can use a more advanced method, such as semantic tableaux. Or we may simply try to find an interpretation under which the formula is false. If there is any one such interpretation, then the formula is invalid.

Example 1.12. Consider the formula $\forall x \exists y Pxy$. Consider the interpretation:

M : the set of people registered for our course
P : ① is taller than ②.

Given this interpretation, the formula says that whatever person in our course you select, there is a person taller than him. Clearly, in our course there will a person who is taller than everyone else in the course. So the formula is false on this interpretation. And therefore, it is invalid.

Similarly, sometimes we wish to know whether a formula is satisfiable—that is, whether it is true under *at least one* interpretation.

Example 1.13. Consider the formula $Pa \supset \forall x Px$. Consider the interpretation:

M : the set of French kings
P : ① is a king
a : Louis XIV.

Clearly the formula is true in this interpretation. Hence it is satisfiable.

Example 1.14. Consider the formula $\forall x(Px \supset Q) \supset \neg \exists x Px$. Consider the interpretation:

M : the set of French kings
P : ① is a king
Q : Moscow is the capital of France.

The formula is true in this interpretation. Hence it is satisfiable. Why?

Chapter 2

Preliminaries

2.1 Philosophical and mathematical motivation

Perhaps one need not any special motivation to study a certain subject. We study logic, one may argue, because it is out there. It seems, however, useful to understand a little more what it is that we shall study, and how the field of logic interacts with other intellectual activities.

We cannot hope to provide a definition of logic. The reason is simple. Logic has been around for more than two thousand years. But even in the past one hundred years activities so multifarious were termed 'logic', that they cannot be covered by a one-line definition. A similar and more familiar problem exists with the definition of mathematics. The issue is interesting philosophically, because a substantive definition may affect the way you think about mathematics, your views on mathematical ontology or mathematical proofs. But clearly, no one-line definition can capture all activities we recognise as 'mathematical'. An interesting attempt was made by the algebraist Serge Lang. He defined mathematics as 'everything published in mathematical journals in the past one hundred and forty years.' Clearly this is not a serious definition (it is circular), but it gives us a hint of how to elucidate the subject of mathematics.

Another hint was given in 1844 by the Russian logician Platon Poretzky who defined formal logic—the kind of logic we are going to study in this course—as logic in its content, but mathematics in its method. Let us, therefore, try to clarify the intended contrast by looking at our logic's historical development, at what was 'published in logical journals in the past two thousand years.' Any such excursus would by necessity contain difficult logical concepts to be refined later in the course; it should be read with care. The earliest European logical treatise belongs to Aristotle. He perceived logic as the science of consequence, determining what follows from what. Why was that important? Suppose the sentence S *entails* the sentence S'. Presumably that means that the truth of S entails the truth of S'. To know such a consequence is useful, as it allows us to know the truth of one sentence just by knowing the truth of another. A question immediately arises as to how we justify our inference. Aristotle collected numerous data on inferences in natural language and assembled them into different forms, the so-called 'syllogistic figures'. They were classified according to their subject-predicate structure. Once the correct forms were established, one could use them for building correct particular arguments. That was the chief utility of logic both for Aristotle and for the Scholastic philosophers of the Middle Ages. The limitations of the Aristotelian treatment have soon become apparent. Syllogistic figures were few in number and involved some very simple sentences. More complex inferences, involving more complex sentences, had to be translated into simpler sentences before any logical analysis of them was possible.

Partly in reaction to these shortcomings a different approach to logic was taken by Leibniz (and some other earlier thinkers, notably by Lull). Leibniz hoped to create a *formalised* universal language. Such a language, using artificial symbols, would have replaced natural language in conducting philosophical and scientific arguments. In the universal language a mathematics of reason will be possible. Every term and every sentence of that language is to be assigned a certain 'character', a number, which will eventually determine the validity of inferences where they occur. Therefore, Leibniz muses, any paradox is really a result of a miscalculation, to be remedied by arithmetical laws of our new grammar. Any philosophical discussion can be solved by computation; instead of engaging in fruitless polemics the opponents should say to each other, 'Let's calculate!'

Example 2.1. Here is an illustration of Leibniz' idea. In modern terms, we investigate the sentences' behaviour

in accordance with their quantificational structure:

Every wise man is righteous.
$+70 - 33 \quad +10 - 3.$

This is a true universal statement. Every term—that is, subject and predicate—is assigned a pair of numbers, each with a different sign. Its truth is reflected in the fact that each number in the characteristic pair of the subject can be divided by a number with the same sign in the characteristic pair of the predicate. When this is not the case, the statement is false.

Formalisation of language received further boost with the work of de Morgan, Boole, and Peirce in the mid-nineteenth century. They were the first to classify inferences according to their sentential structure. But it is only with Frege that logic begins to play a novel role as the language of mathematics. In fact, Frege attempted to reduce all mathematics to logic.

Another crucial step in the development of logic was the emergence of formal axiomatic systems, where major contribution was made by Hilbert. When mathematical theories were represented in the form of axiomatic systems, it became possible to explore the properties of those systems. The properties in question, *e.g.* consistency, were logical.

Since then foundational issues, justification of the mathematical discipline as a whole, that animated Frege and Hilbert, have lost their appeal for mathematicians. They remain popular with some philosophers. But the methods of model theory, an offspring of the axiomatic approach, have proved exceptionally fruitful in investigating the properties of various mathematical theories, notably set theory, abstract algebra, geometry, and analysis.

What is the significance of formal logic for philosophers? First of all, many issues in the philosophy of language, such as the theory of meaning and theory of reference have direct connection to the systematic formalisation of natural language. They become inseparable from a deep analysis of logical constants, quantifiers, and predicates that can be conducted rigourously with the machinery of first-order logic. Secondly, many metaphysical and epistemological topics, such as conceptual knowledge or analyticity, are best explored by formal tools. (*Cf.* Dummett's remark...) Then there is the philosophical import of Gödel's theorems, the nature of Turing machines, and their impact on the theory of truth, the conception of rationality, the mind/body problem, and so forth.

That said, there is no need to constantly question the utility of logic for specific philosophical and mathematical concerns, the scope of its application in solving various puzzles. Our interest in it must be pure and selfless, much like our interest in philosophy or mathematics.

2.2 General concepts

Let us introduce some concepts which will be of use in the future. We presuppose familiarity with the notation of sentence logic. The signs $\wedge$, $\vee$, $\neg$, $\rightarrow$ will stand for conjunction, disjunction, negation, and implication respectively. One should note that our logical symbols are used as meta-linguistic abbreviations, that is, they are not part of the language of the set theory we develop. (In subsequent chapters we shall employ a different sign for implication.)

2.2.1 Sets

The concept of a set can be thought of as a generalisation of the concept of collection. Any collection of objects will constitute a set. The fact that a certain object x is *included* in the set Z will be symbolised as $x \in Z$.[1] The notion of inclusion is a primitive notion. Sets are identified by the objects they include. Thus, the *axiom of extensionality* states that if any two sets include the same objects, they are identical. That is, to specify a set it suffices to specify all of its elements.

The notion of inclusion allows us to define some further set-theoretic operations as follows:

- A set X is a subset of a set Y—symbolically, $X \subseteq Y$—if every element included in X is also included in Y. The empty set $\varnothing$ includes no elements at all. By those definitions we conclude that for any set X, $X \subseteq X$ and $\varnothing \subseteq X$.
- A set X is a *proper subset* of Y—symbolically, $X \subset Y$—just in case $X \subseteq Y$ and $X \neq Y$.

[1] The sign $\in$ derives from the Greek verb *esti*, 'to be.'

- The *union* of X and Y is the set $Z = X \cup Y$ which includes all the elements of X and Y.
- The *intersection* $Z = X \cap Y$ is the set including the elements belonging both to X and Y.
- The *difference* $Z = X - Y$ is the set including the elements of X which are not in Y.

When $Y \subseteq X$, it is also useful to introduce the *complement* of Y in X which is just the difference $X - Y$. We shall designate the complement of Y as Y'.

Set-theoretic operations can be usefully explained with the aid of Venn's diagrams. The idea is to take as the basic set (the universe set) the collection of points on the plane. We represent the sets X and Y as circles (that is, containing points within their circumferences). Then the operations just discussed can be depicted as in Figure 2.1 where they are represented by hatched regions (the definition of $X \oplus Y$ is left as an exercise).

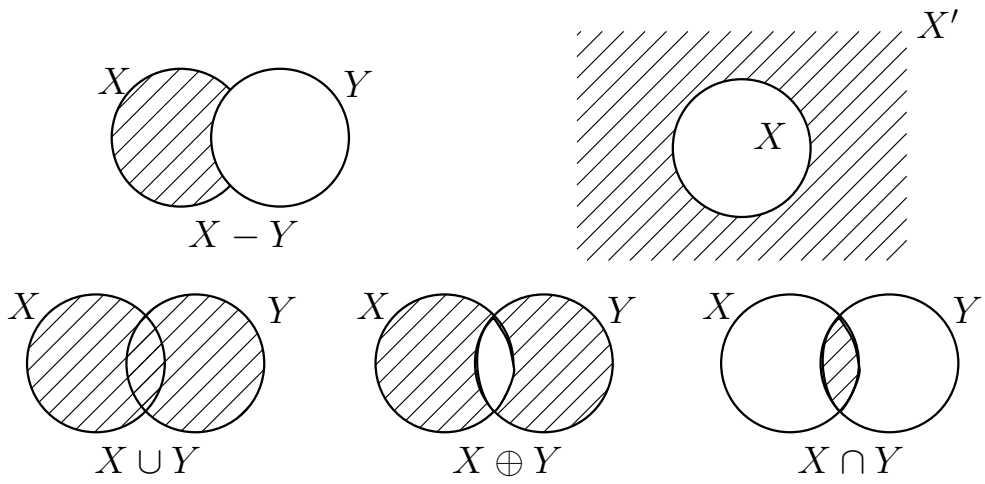


Figure 2.1: Set-theoretic operations

We now move on to introduce relations and mappings. The *cartesian product* $Z = X \times Y$ of the sets X and Y is the collection of all pairs $\langle a, b\rangle$ such that $a \in X$ and $b \in Y$. Note that $X \times Y$ is not equal to $Y \times X$. If $A_1 = \cdots = A_n$, then the cartesian product $A_1 \times \cdots \times A_n$ will be designated as A^n. Cartesian product

Example 2.2. Let $X = \{7, 8\}$ and $Y = \{5, 6\}$. Then $Z = X \times Y = \{\langle 7, 5\rangle, \langle 7, 6\rangle, \langle 8, 5\rangle, \langle 8, 6\rangle\}$.

One way of defining binary relations is to draw their graphs on the Cartesian plane.

Example 2.3. Consider $\alpha = \{\langle x, y\rangle \mid x, y \in \mathbb{R} \wedge x > 2 \wedge y < 3\}$.

Example 2.4. Consider $\alpha = \{\langle x, y\rangle \mid x, y \in \mathbb{R} \wedge x = y\}$.

Each subset α of $X \times Y$ is called a relation on X and Y. If the pair $\langle a, b\rangle \in \alpha$, then we shall say that a stands in relation α to b. We can introduce the following operations on relations:

$$a(\alpha \cup \beta)b \iff a\alpha b \text{ or } a\beta b;$$
$$a(\alpha \cap \beta)b \iff a\alpha b \text{ and } a\beta b;$$
$$a\alpha' b \iff \text{not } a\alpha b.$$

We may therefore speak about disjunction, conjunction, and negation of relations.

Example 2.5. The relation $=$ of equality in the set $\mathbb{N}$ of natural numbers can be thought of as a collection of all pairs $\langle 0, 0\rangle, \langle 1, 1\rangle, \ldots$. The complement of this relation is the inequality relation. The relation $<$ is the set of all pairs $\langle a, b\rangle$ such that $a < b$. The relation $\leq$ is the same as the relation $< \cup =$. The relation $< \cap =$ is empty—that is, it is a necessarily false relation. By contrast, the relation $\leq \cup >$ is necessarily true.

Two more operations on relations should be mentioned. The *inverse* of α is the relation α^{-1} such that $\langle b, a\rangle \in \alpha^{-1}$ just in case $\langle a, b\rangle \in \alpha$. Suppose now that α is a relation of X to Y and β is a relation of Y to Z. The *composition* $\alpha\beta$ of X to Z is a relation such that $\langle a, b\rangle \in \alpha\beta$ just in case there is an element x such that $a\alpha x$ and $x\beta b$. The relations α and β are *permutable* if $\alpha\beta = \beta\alpha$. The *identity relation* in a set X consists of all pairs $\langle a, a\rangle$, where $a \in X$, and is denoted by ι_X.

2.2.2 Mappings

Among various kinds of relations several have particular importance.

Definition 2.6. A relation α defined on X and Y is called a *mapping* of X into Y if for each $a \in X$ there is exactly one $b \in Y$ such that $a\alpha b$. The element $b = a\alpha$ is the *image* of a, and a is the pre-image of b.

Definition 2.7. The set of all x for which there is y such that $x\alpha y$ is called the *domain* of α. The set of all y for which there is x such that $x\alpha y$ is called the *range* of α.

The fact that α is a mapping of X into Y is designated as $\alpha\colon X \to Y$. The fact that $b = a\alpha$ may sometimes be convenient to designate as $\alpha\colon a \mapsto b$. One may verify that a familiar notion of function, as understood, *e.g.*, in analysis, coincides with the notion of mapping. An n-ary *operation* is the mapping of A^n into A.

Definition 2.8. A mapping α of X into Y is called a *mapping of X onto Y* if for each $b \in Y$ there is at least one $x \in X$ such that $x\alpha b$. A mapping α of X onto Y is a *bijection* (or *one-to-one*) if the inverse relation α^{-1} is a mapping of Y onto X. Bijection

Remark. The definition of mapping-into guarantees that each element in the range of a mapping-onto has *exactly one* pre-image.

Example 2.9. The function $f(x) = 4x$ defined on reals is a bijection. The function $f(x) = x^2$ defined on reals is not a bijection. why? ...

Consider now a bijection α of X onto Y. Its inverse α^{-1} is also a bijection. Since for any $x \in X$ and $y \in Y$ we have $(x\alpha)\alpha^{-1} = x$ and $(y\alpha)\alpha^{-1} = y$, it follows that $\alpha\alpha^{-1} = \iota_X$ and $\alpha^{-1}\alpha = \iota_Y$. When α is a bijection of X onto X, then $\alpha\alpha^{-1} = \alpha^{-1}\alpha$. This in fact gives us a necessary and sufficient condition for a relation to be a bijection.

The notion of bijection allows us to introduce cardinalities. With each set X we associate a *cardinal* (or a *cardinal number*, or *cardinality*, or *power*) denoted by $|X|$ such that $|X| = |Y|$ just in case there is a bijection between X and Y. Thus, the empty set $\varnothing$ is assigned the cardinal 0, while the set $\{x_1, \ldots, x_n\}$ is assigned the cardinal n. The cardinal of the set of all natural numbers is denoted by $\aleph_0$. Where $|X| = |Y|$ the sets X and Y are called *equinumerous*. Further, if there is a one-to-one mapping of X into Y, then $|X| \leq |Y|$. A set X is *finite* if it is either empty, or else equinumerous with the set $\{1, 2, \ldots, n\}$ for some natural number n. Sets equinumerous with the set of all natural numbers are called *denumerable*. Sets which are either finite or denumerable are called *countable*. Equinumerousity of X and Y is denoted by $X \approx Y$. Cardinality

Example 2.10. The sets $X = \{0, 1, 2, 3, \ldots\} = \mathbb{N}$ and $Y = \{0, 2, 3, 4, \ldots\} = \mathbb{N} - \{1\}$ are equinumerous, since it is possible to find a bijective mapping of X into Y.

2.2.3 Equivalence relations

We shall now introduce a further important class of relations.

Definition 2.11. A binary relation α in a set X is an *equivalence relation* on X if it satisfies the following conditions: Equivalence relation

- Reflexivity: $x\alpha x$,
- Symmetry: if $x\alpha y$, then $y\alpha x$,
- Transitivity: if $x\alpha y$ and $y\alpha z$, then $x\alpha z$,

for all $x, y, z \in X$.

Our discussion above allows to express the properties of equivalence relations also in the following form: why? ...

- Reflexivity: $\iota \subseteq \alpha$;
- Symmetry: $\alpha^{-1} \subseteq \alpha$;
- Transitivity: $\alpha^2 \subseteq \alpha$.

A family M of non-empty subsets of a set X is called a *partition* of X if each element of X belongs exactly to one element of M. The elements of M are called *classes* of the partition.

Example 2.12. Let R be the set of pairs $\langle x, y\rangle$ such that x and y are lines on the Euclidean plane and x is parallel to y. Then R is an equivalence relation.

Example 2.13. The set $X = \{x, y\}$ has two partitions: $M_1 = \{\{x, y\}\}$ and $M_2 = \{\{x\}, \{y\}\}$. The set $X = \{x, y, z\}$ has five partitions: $M_1 = \{\{x, y, z\}\}$, $M_2 = \{\{x\}, \{y\}, \{z\}\}$, $M_3 = \{\{x\}, \{y, z\}\}$, $M_4 = \{\{x, z\}, \{y\}\}$, $M_5 = \{\{x, y\}, \{z\}\}$. See Figures 2.2 and 2.3.

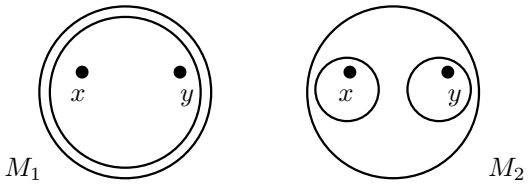


Figure 2.2: Partition for $\{x, y\}$

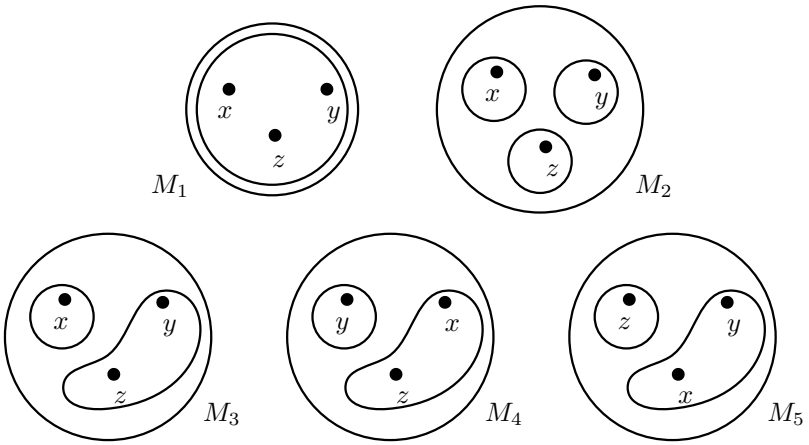


Figure 2.3: Partition for $\{x, y, z\}$

With each partition M of the set X we can associate a relation ϕ such that for any $x, y \in X$, $x\phi y$ just in case x and y belong to the same class of partition. Clearly the relation ϕ is an `equivalence relation`. We say that ϕ is *induced* by M. why? by definition

We can show that each equivalence relation ϕ induces a partition. For each $x \in X$, let $[x]$ be the set of all elements u such that $u\phi x$. The sets $[x]$ are called *equivalence classes* (see Figure 2.4). We further note that

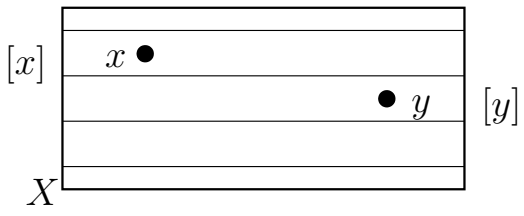


Figure 2.4: Equivalence classes

since ϕ is reflexive, $x \in [x]$. We must show that if $y \in [x]$, then $[y] = [x]$:

$y \in [x]$	Ass.	(1)
$y\phi x$	(1), def. of equivalence classes	(2)
$u \in [x]$	Ass.	(3)
$u\phi x$	(3), def. of equivalence classes	(4)
$x\phi y$	(2), ϕ is symmetrical	(5)
$u\phi y$	(4), (5), ϕ is transitive	(6)
$u \in [y]$	(6)	(7)
$[x] \subseteq [y]$	(3), (7)	(8)
$u \in [y]$	Ass.	(9)
$u\phi y$	(9), def. of equivalence classes	(10)
$u\phi x$	(9), (2), ϕ is transitive	(11)
$[y] \subseteq [x]$	(9), (11)	(12)
$[x] = [y]$	(8), (12).	(13)

Therefore, the family of all equivalence classes induced by ϕ is a partition of X. This family is called the *factor set of X by ϕ* and is designated as X/ϕ. We have shown that $x\phi y$ just in case there is a set $M \in X/\phi$ such that $x \in M$ and $y \in M$.

2.2.4 Trees

An *unordered tree* $\mathscr{T}$ is a triple $\langle S, \ell, R\rangle$ consisting of the following:

1. A set S of *points*;
2. A function ℓ assigning to each $x \in S$ a natural number $\ell(x)$ called the *level* of x;
3. A binary relation ①R② defined in S which is interpreted as '① is a *predecessor* of ②' or '② is a *successor* of ①'. This relation satisfies the following conditions:

   C_1: There is a unique point a_1 of level 1 called the *origin* of the tree;

   C_2: Every point save for the origin has a unique predecessor;

   C_3: For any $x, y \in S$, if xRy, then $\ell(y) = \ell(x) + 1$.

A point x is an *end point* if it has no successors; it is a *simple point* if it has just one successor; and it is a *junction point* if it has more than one successor. A *path* is any ordered sequence of points containing the origin such that each of its terms is the predecessor of the next (that is, except the last one, if there is such). A *branch* is a path whose last term is an end of the tree, or else it is an infinite path.

The conditions on R mean that that for any $x \in S$, there exists a unique path P_x whose last term is x. If y lies on P_x, then y *dominates* x. In that case, and if $x \neq y$, then y lies *above* x. The points x and y are *comparable* if one of them dominates the other. The point y lies *between* x and z if y is above one element of the pair $\langle x, z\rangle$ and below the other.

An *ordered tree* $\mathscr{T}$ is a quadruple $\langle S, \ell, R, \theta\rangle$, where θ is a function assigning to each point x an ordered sequence $\theta(x)$ containing all the successors of x. We shall then be able to speak about the first, second, and so on, successors of x.

Sometimes we shall have to add new points to the given ordered tree $\mathscr{T} = \langle S, \ell, R, \theta\rangle$. This will be done by adding them as successors to end points. Namely, to add distinct elements $y_1, y_2, \ldots, y_n$ to an end point $x \in S$ we add each y_i to S, add each $\langle x, y_i\rangle$ to R, let $\ell(y_1) = \ell(y_2) = \cdots = \ell(y_n) = \ell(x) + 1$, and let $\theta(x) = \langle y_1, \ldots, y_n\rangle$.

Most frequently we shall use *dyadic* ordered trees, in which each junction point has exactly two successors. The first successor will be called the *left successor*, and the second will be called the *right successor*.

A tree is called *finitely generated* if each point has a finite number of successors. A tree is *finite* if it has only finitely many points. A tree is *infinite* if it has infinitely many points. Note that a finitely generated tree may be infinite.

Example 2.14. The abstract notion of a tree we have given has a simple graphical representation. Consider a dyadic tree $\mathscr{T} = \langle S, \ell, R, \theta \rangle$, where we let $S = \{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$, $\ell(\mathbf{A}) = 1$, $\ell(\mathbf{B}) = 2$, $\ell(\mathbf{C}) = \ell(\mathbf{D}) = 3$, and $\theta(\mathbf{B}) = \langle \mathbf{D}, \mathbf{C} \rangle$. We draw the tree by placing the origin at the top, the successor y of each point x below x, and connect x and y with a line. The successors are ordered from left to right. The resulting tree is shown on Figure 2.5.

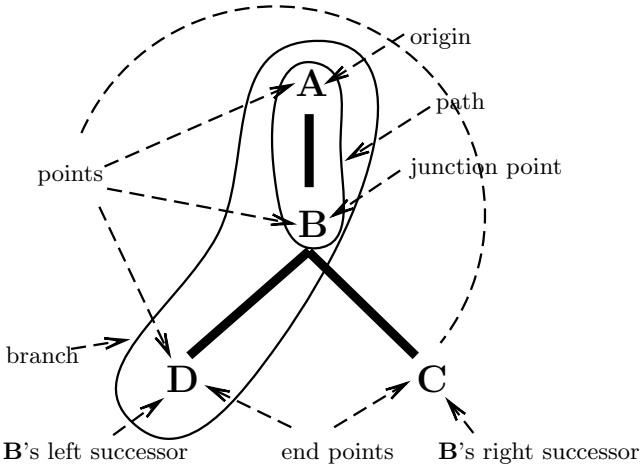


Figure 2.5: A dyadic tree

2.3 Logic and metamathematics

2.3.1 Axiomatic systems

The axiomatic method in mathematics goes back to Euclid's treatise on geometry. It can be characterised as follows. At the beginning we introduce primitive terms with the aid of definitions. Examples will include 'point', 'straight line', 'plane'. Definitions establish the meaning of the terms. We then formulate propositions involving those terms whose truth is seen as intuitive and in no need of proof. The source of their truth is supposed to lie in the very meaning of the terms. Euclid differentiates between *postulates* and *axioms*, but we may agree on calling all such propositions 'axioms'. Using primitive terms we may define new terms, and from the axioms logically derive new propositions, the *theorems* of our axiomatic system.

Such a system, in which the meaning of primitive terms is fixed from the start, may be termed 'informal'. Historically, the impetus for developing an alternative conception of axiomatic system, a 'formal' one, was given by the continuing uncertainty over Euclid's Fifth Postulate. The principal role of that postulate was in proving the theorem that through a point P not lying on a given line l it is possible to draw exactly one line parallel to the original line l. Both the original postulate and the theorem are not quite intuitive. How would one know, for instance, that the two allegedly parallel lines do not meet in some distant point in space?

A lot of effort was invested in proving the Fifth Postulate from the rest of the axioms. All these attempts were destined to fail, since in the early 19th century there was built a system in which through the point P, described as above, it is possible to draw infinitely many lines parallel to l.

Whereas in the past, prior to the 19th century, the axioms were required to be self-evident, there is no such demand in the modern conception of axiomatic systems. Arbitrary sentence may serve as axioms. This view is directly linked to the emergence of non-Euclidean geometries.

2.3.2 Language and metalanguage

Suppose we have an axiomatic system. We wish to investigate its properties. Then, on the one hand, we have a language in which our system is expressed, and, on the other hand, a language in which we talk about it. Those two languages do not need to coincide.

This is a perfectly familiar situation. A Turkish language textbook for English speakers will have Turkish as object-language and English as metalanguage. We similarly will study formal languages with the aid of English as the metalanguage.

The distinction generalises into the use-mention distinction. Consider the statement:

Ankara is pretty.

Following Quine, we say that the name 'Ankara' is *used* here to *mention* the city named by it. If, however, we write:

'Ankara' is pretty,

then we express our admiration for the name, not for the city. That is to say, the name is mentioned, whereas the quotation is used to refer to the name. Quotation-marks, then, are a device for jumping the level in the use-mention hierarchy.

For our later discussion it will be important to keep in mind the distinction between variables and meta-variables. Variables typically take as their values different objects. For examples:

Variables and meta-variables

Let there be two people, x and y, and x loves y.

Meta-variables act just like variables, except that they are designed to occur in the meta-language. Since meta-language is used to make claims about an object-language, it is to be expected that the values of meta-variables will typically be linguistic items—which themselves belong to the object-language. For example:

Let A be a formula of the Euclidean geometry.

Here the letter 'A' is used as a name of some formula of the Euclidean geometry. The formula itself belongs to the object-language, *i.e.* the language of the Euclidean geometry. The letter 'A' belongs to the meta-language and should be regarded as a meta-variable. (Compare: 'Let 'Tibbs' be the name of our cat.')

2.3.3 Set-theoretic constructions

Every discipline pays at least attention to its own language. But the focus on language in mathematics is unique. This can be partly explained by essential epistemological problems. In natural science the object of investigation is given. There is no need to doubt its reality unless one is engaged in a global sceptical enquiry. In mathematics the object of investigation is elusive. It is not clear in what sense numbers and sets are real. It is similarly unclear, for the same reason precisely, in what sense mathematical claims can be true.

This difficulty has important consequences. In building the edifice of formal logic we must use intuitive, informal notions. Metalanguage is their medium. Also, the focus on truth prompts deep enquiries into its behaviour in mathematical theories. They result in claims about incompleteness, independence, undecidability of various axiomatic systems.

Various schools in philmath....

Developments in set theory had considerable influence on the methods of modern logic. We give several illustrations.

Cantor's diagonal method

Proposition 2.15 (Cantor)**.** *There is no 1-1 mapping between the set* $\mathbb{N}$ *and the set* $A = \{x \in \mathbb{R} \mid 0 < x < 1\}$.

Proof. We represent the members of A in the form $0, \alpha_1\alpha_2 \cdots \alpha_n \cdots$, where α_i is a decimal digit.

Suppose that A is denumerable. Then there is a bijection between $\mathbb{N}$ and A, and we can enumerate the members of A. We have:

$$\begin{aligned} a_1 &= 0, \alpha_{11}\alpha_{12} \cdots \alpha_{1j} \cdots \\ a_2 &= 0, \alpha_{21}\alpha_{22} \cdots \alpha_{2j} \cdots \\ &\cdots \\ a_i &= 0, \alpha_{i1}\alpha_{i2} \cdots \alpha_{ij} \cdots \\ &\cdots \end{aligned}$$

For each $j \in \mathbb{N}$ we can select a decimal digit β_j such that $\beta_j \neq \alpha_{jj}$, and β_j is not either a the digit '9' or '0'. Consider the number:

$$b = 0, \beta_1\beta_2 \cdots \beta_j \cdots .$$

Now $b \in \mathbb{R}$ and $0 < b < 1$. Therefore, there is $i \in \mathbb{N}$ for which $b = a_i$. Then $\beta_j = \alpha_{ij}$ for every $j \in \mathbb{N}$. In particular, $\beta_i = \alpha_{ii}$. But this is impossible, since we selected b precisely to satisfy the condition $\beta_i \neq \alpha_{ii}$. We obtained a contradiction. Therefore, A is not denumerable. □

Russell's Paradox

Definition 2.16. A set X is *good* if it is not an element of itself. X is *bad* otherwise.

Consider the set Σ that contains all the good sets. Is Σ good or bad?

1. Suppose that Σ is good. Then Σ must contain itself by the definition of Σ. But then Σ is bad by the definition of badness. Contradiction.
2. Suppose that Σ is bad. Then Σ must not contain itself by the definition of Σ. But then Σ is good by the definition of goodness. Contradiction.

A natural conclusion of this reasoning is that Σ does not exist.

Grelling's Paradox

Definition 2.17. An adjective E of the English language is called 'autological' if E describes itself. E is *heterological* otherwise.

So, for instance, 'short' is autological, but 'long' is heterological. Now is 'heterological' heterological?

Berry's Paradox Each of the English names of natural numbers contains a number of words. For example, the first number requiring at least two words is 21. Let us consider the name (or more exactly, the definite description) of a certain number: 'The first number whose name requires at least eleven words.' This is an adequate name (description) of a number. Yet it contains just ten words.

2.3.4 The universal compiler

We shall now see how the methods just described may be applied to a logical problem. It is related with Gödel's incompleteness results.

Suppose that a computing centre receives texts written in a programming language, such as PASCAL. A special program translates these texts into programmes written in machine code. Some texts contain errors (e.g. syntactic errors), and the translator is made to react to them by halting.

A question arises if it is possible to have a translator which would discover *all* errors in any input text.

To fix the terminology, we consider a programming language L. Its alphabet contains a finite number of symbols. Any finite sequence of those symbols will be a *text*. The text is a 'correct program' if it is possible to translate it into machine code and if the translated code allows the computer to work for infinitely long time printing as its output the table of the form:

$$\begin{array}{ccccc} 1 & 2 & 3 & 4 & \cdots \\ f(1) & f(2) & f(3) & f(4) & \cdots \end{array}$$

where $f(n) \in \mathbb{N}$.

Definition 2.18. The program COMPILER is a program whose input is any text and whose output is 1 if the text is a correct programme, and 0 otherwise.

Proposition 2.19. *The* COMPILER *does not exist.*

Proof. Suppose that the COMPILER exists. We shall now write a certain Big Program which will work as follows.

1. Arrange all symbols of L in alphabetic order.
2. A special subroutine of BP generates all the texts of L arranged by length, while the texts of the same length are arranged in lexicographical order.
3. A text is received on the input of the COMPILER. If the output is 0, this text is discarded. If the output is 1, then the text is a correct program, and it is assigned the index i. A subsequent correct program gets the index $i + 1$. In this way we generate all and only all correct programs in lexicographic order.
4. Let the nth correct program compute the function $f_n(k)$. The final subroutine of BP computes for every $n \in \mathbb{N}$ the number $F(n) = f_n(n) + 1$. It generates the pair $(n, F(n))$ in the output.

Now the program BP is itself a correct program. So the function $F(n)$ must be contained in the array $\langle f_1, f_2, \ldots, f_m, \ldots \rangle$ of *all* the functions computed by correct programs.

But this is impossible. F cannot be in the array, since $F(n) \neq f_n(n)$ for every $n \in \mathbb{N}$. □

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 $\neg A$;

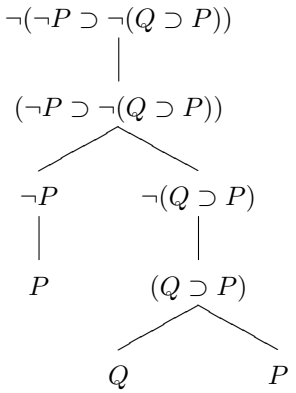
FR3. If A and B are formulae of $\mathsf{H_s}$, then so is $(A \supset B)$.

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:

Q	by FR1
P	by FR1
$(Q \supset P)$	by FR3
$\neg(Q \supset P)$	by FR2
$\neg P$	by FR2
$(\neg P \supset \neg(Q \supset P))$	by FR3
$\neg(\neg P \supset \neg(Q \supset P))$	by FR2.

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



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:

$$\begin{aligned}\text{Evil-doer} &= \text{dragon}\\ \text{Dwelling-place} &= \text{residence}\\ \text{Mail-shirt} &= \text{armour}\\ \text{Folk-right} &= \text{possession}\\ \text{Stone-cliff} &= \text{wall}\end{aligned}$$

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.

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

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 $A \supset C_i$:

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

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 $A \supset C_i$:

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

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

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

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

$$(A \supset (C_k \supset C_i)) \supset (A \supset C_k \supset (A \supset C_i))$$
$$A \supset C_k \supset (A \supset C_i).$$

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 **1** and **0** are primitive elements. We can conduct the procedure in a systematic way. Sentence parameters standing for true sentences will be assigned the value **1**, and those standing for false sentences will be assigned the value **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 **1** and **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 **0** and **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.*

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. □

Induction on the complexity of a formula

Let us now define two other important notions.

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

Satisfiability

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*.

Truth sets

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 **0** in its last column, but where all sentence parameters are assigned the value **1**.

To do this rigorously, we must first show that every formula A of the language L is assigned **1** when all of its sentence parameters are assigned **1**. We can induce on the complexity of A. Let V be such a valuation of the signature Σ that it assigns **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 **1** for all sentence parameters and the value **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. □

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 $\mathscr{T}_1$.

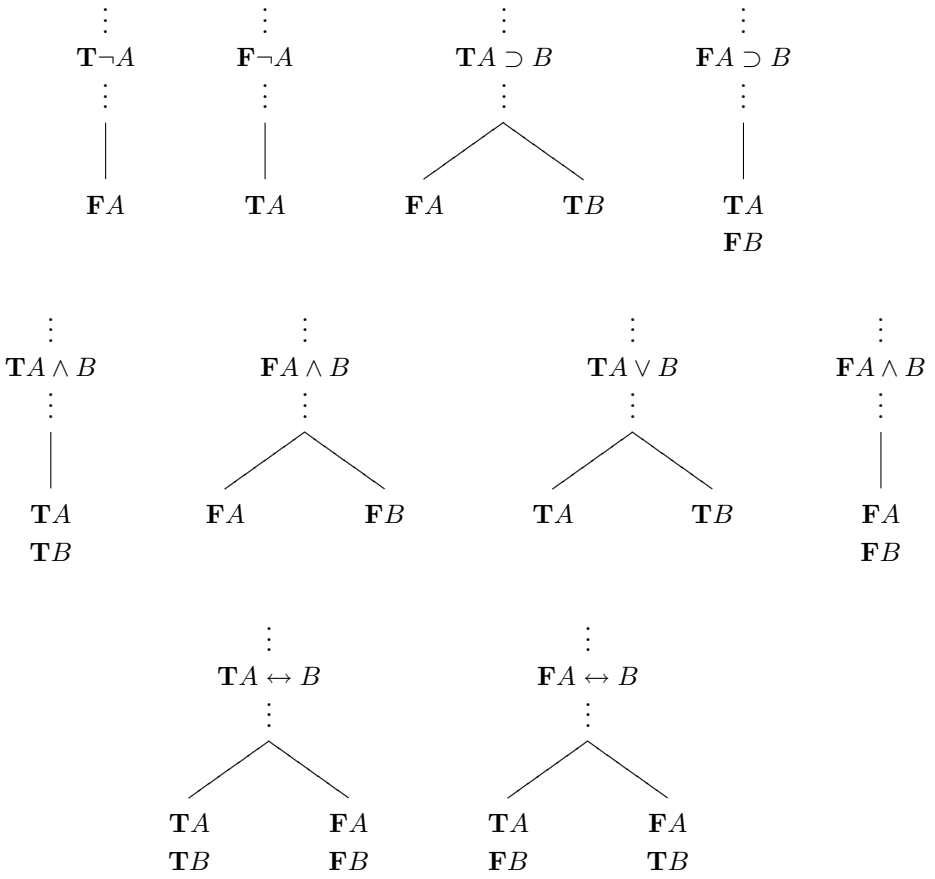


Table 4.1: Signed tableau processing rules

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 $\mathscr{T}_i$.

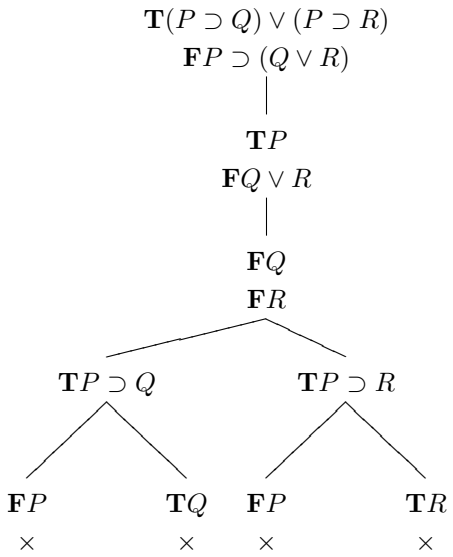
Definition 4.29. A branch θ of a tableau $\mathscr{T}$ is *closed* if it contains a conjugate pair of formulae. We shall indicate that θ is closed by inserting a cross $\times$ into it. The tableau $\mathscr{T}$ is closed if all of its branches are closed.

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 $(P \supset Q) \vee (P \supset R) \vdash P \supset (Q \vee R)$:



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.

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.

Properties of truth sets

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 $\Gamma_1, \Gamma_2, \ldots$ 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 $\Delta' = \{A_1, \ldots, A_n\} \subseteq \Delta$. 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 $f(1), \ldots, f(n)$. 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 $A_1, \ldots, A_n$ be an ordering of all the formulae of Σ. We define the sequence $\Gamma_1, \Gamma_2, \ldots$ 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 $\Gamma \vDash A$. □

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 **??.??**. □

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. □

4.6 Gentzen systems

To be omitted. May be introduced at the end.
[Bos97, Men64, Smu68]

Chapter 5

Predicate calculus: Syntax

In this chapter we begin investigating the properties of predicate calculus. We shall see that, while some concepts and results of propositional calculus straightforwardly carry over to the case of predicate calculus, still many other are significantly generalised or else are completely novel.

5.1 Quantifiers and variables

It is clear that the resources of propositional calculus are inadequate for identifying valid arguments in many areas of discourse. The inference 'If Socrates is male, then he is not female' appears to be a valid one. But all we can do within the limits of $\mathsf{H_s}$ is to symbolise it as 'If P, then Q', or perhaps as 'If P, then $\neg Q$', but in either case there is no reason why these arguments must be recognised as valid. The apparent validity of our inference derives from a link between Socrates, the property of being female, and the property of being male. That link remains inscrutable so far as we are confined to $\mathsf{H_s}$. Thus we must enrich our syntactic and semantic means to talk about objects and relations (recall that properties are unary relations). This will be achieved in predicate calculus.

Some of the elements of predicate calculus should be familiar from the introductory course. Here we start by drawing attention to just one specific issue of philosophical, rather than mathematical, significance—namely, the role of variables—and proceed straightaway to the formal exposition.

The true claim 'The number 9 is less than, or equal, or greater than 0' can be paraphrased as a disjunction:

$$(9 < 0) \vee (9 = 0) \vee (9 > 0).$$

To write this disjunction in our canonical propositional notation we may simply treat each disjunct as an atomic sentence denoting it by a sentence parameter. But consider another true statement, 'Every real number is less than, or equal to, or greater than 0'. Here an attempt to ape our disjunction:

$$(\text{Every real number} < 0) \vee (\text{Every real number} = 0) \vee (\text{Every real number} > 0)$$

fails, since this paraphrase is obviously false. Rather, we have to labour a bit, first representing the original statements as:

Whatever real number is selected, it is either less than, or equal to, or greater than 0.

Therefore:

Whatever real number is selected, $((\text{it} < 0) \vee (\text{it} = 0) \vee (\text{it} > 0))$.

However, suppose we advance further and wish to make a claim previously made about 0 about every number. That will be another true claim, since, mathematically, there is nothing unique about 0 in this respect. Then we may say:

Whatever real number is selected, (every real number is either less than it, or equal to it, or greater than it).

But the parenthetical expression still demands a paraphrase. According to our previous strategy, we may expect the following:

Whatever real number is selected, (whatever real number is selected, $((\text{it} < \text{it}) \vee (\text{it} = \text{it}) \vee (\text{it} > \text{it}))$).

This will not do: unless we separate between two selections—that is, two occurrences of 'whatever' in our statement—we will be making a false claim. To distinguish between them, we may resort to indexing. We shall attach the same indices to 'whatever' and to the locutions of 'it' fixed by 'whatever'. Accordingly:

$$\text{Whatever}_1 \text{ real number is selected, (whatever}_2 \text{ real number is selected, ((it}_2 < \text{it}_1) \vee (\text{it}_2 = \text{it}_1) \vee (\text{it}_2 > \text{it}_1))).$$

But instead of indexing it would be more convenient to distinguish the *scope* of the occurrence of 'whatever' by different letters. Thus, the latest paraphrase will take the form:

$$\text{Whatever real number } x \text{ is selected, (whatever real number } y \text{ is selected, } ((y < x) \vee (y = x) \vee (y > x))).$$

On the other hand, the locution 'whatever entity is selected' is synonymous with the locution 'for all entities'. We further abbreviate 'for all' as '$\forall$'. Yet the paraphrase:

$$\forall x(\forall y((y > x) \vee (y = x) \vee (y < x))).$$

would be too quick: we have omitted the provision made for real numbers. The remedy is not difficult: the claim 'For all Fs, $\cdots$' may be paraphrased as the claim 'For all x, if x is F, then $\cdots$'. By inserting the conditional we avoid making claims about any non-F. Hence the correct paraphrase would be as follows:

$$\forall x(x \text{ is a real number } \supset (\forall y(y \text{ is a real number } \supset ((y > x) \vee (y = x) \vee (y < x))))).$$

We shall have an opportunity to explore the properties of quantifiers in more detail later on. What is important right now is to notice that the variables as introduced by us have the sole role of cross-referencing the quantifier. The indexing tool we abandoned for the purpose of convenience displayed just that. In this way variables are distinguished from names, or 'individual constants'. This may seem obvious: the whole purpose of moving from the specific claims about 9 and 0 above was in eliminating the names for these numbers in our statements. None the less it is not uncommon to encounter claims, such as 'Take any number—say, 22', or 'Take any politician—say, Tony Blair'. The speaker may go on and predicate properties of the number 22 or Tony Blair. Thus, if the resulting claim has the form 'x is F', x would appear to be a proper name—either for 22, or Tony Blair. Such a practice would be *logically* fallacious for a number of reasons. The simplest fallacy, to cut the story short, is that the original intention of the speaker was to reason about properties of *every* number or politician, whereas 22 and Tony Blair may not share properties with every number or politician. In general, the use of variables allows us making claims about *arbitrary* entities, but it does not validate replacing those variables by names of specific entities. (In daily contexts, the speaker is likely to be interpreted as making an inductive inference: by examining the properties of Tony Blair and, say, Jacques Chirac, he leaps to a claim about politicians in general.)

5.2 First-order theory

We shall present the system $\mathsf{T_p}$ of predicate calculus. To begin with, the above discussion suggests a revision of our concept of signature. We formulate it in the most general way, and will simplify later.

Definition 5.1. Let Σ_v be a denumerable set of individual variables. Let Σ_{ip} be a denumerable set of individual parameters. Let Σ_c be a set of individual constants. Let Σ_p be a denumerable set containing i-ary predicate parameters for each $i \geq 0$. Let Σ_f be a denumerable set containing i-ary function parameters for each $i \geq 0$. The *signature* for $\mathsf{T_p}$ is the set $\Sigma = \Sigma_v \cup \Sigma_{ip} \cup \Sigma_c \cup \Sigma_p \cup \Sigma_f$.

Remark. The sets Σ_v, Σ_{ip}, Σ_c, Σ_p, and Σ_f are mutually disjoint.

Predicate parameters having the arity n should be thought as standing for the sentences in the object language that have n blanks. When we wish to identify the predicate, those blanks will be represented by circled numerals (see Exercises for an example). Consequently, an 0-ary predicate parameter will be nothing but a sentence parameter. Similarly, constants can be thought as 0-ary function parameters, as we have mentioned already whilst discussing tautologies and contradictions. On occasions, it is convenient to specify the arity of predicates (and functions) explicitly. For an i-ary predicate P_j we shall then write 'P_j^i'.

The sign $\forall$ will stand for the universal quantifier. The existential quantifier will be introduced as an abbreviation. We can now define terms and formulae of our theory as follows.

Definition 5.2. The set $\mathbf{T}_\Sigma$ of terms of the signature Σ for $\mathsf{T_p}$ is the smallest set of expressions determined as follows:

1. $\Sigma_v \cup \Sigma_{ip} \cup \Sigma_c \subseteq \mathbf{T}_\Sigma$;
2. If $t_1, \ldots, t_n \in \mathbf{T}_\Sigma$ and $f \in \Sigma_f$, then $f(t_1, \ldots, t_n) \in \mathbf{T}_\Sigma$.

Definition 5.3. The set $\mathbf{F}_\Sigma$ of formulae of the signature Σ for $\mathsf{T_p}$ is then determined by the following rules:

1. If $P \in \Sigma_p$ is an 0-ary predicate parameter, then $P \in \mathbf{F}_\Sigma$.
2. If $P \in \Sigma_p$ is an n-ary predicate parameter, where $n > 0$, and $t_1, \ldots, t_n \in \mathbf{T}_\Sigma$, then $Pt_1 \cdots t_n \in \mathbf{F}_\Sigma$.
3. If $A \in \mathbf{F}_\Sigma$, then $\neg A \in \mathbf{F}_\Sigma$.
4. If $A, B \in \mathbf{F}_\Sigma$, then $(A \supset B) \in \mathbf{F}_\Sigma$.
5. If $A \in \mathbf{F}_\Sigma$, $x \in \Sigma_v$, then $\forall x A \in \mathbf{F}_\Sigma$.

Remark. Unless explicitly indicated to the contrary, we shall assume that $\Sigma_f = \Sigma_c = \varnothing$. Note also that each n-ary function $f\colon X \to Y$ can be 'represented' by an $(n+1)$-ary predicate as follows:

$$f(x_1, \ldots, x_n) = y \iff P(x_1, \ldots, x_n, y),$$

where $x_1, \ldots, x_n \in X$, $y \in Y$. This will allow us talking about various mathematical structures without using function parameters.

Definition 5.4. Let $\mathbf{Sm}_\Sigma = \Sigma \cup \{\neg, \supset, \forall, (,)\}$. Then $\mathbf{Sm}_\Sigma$ is the set of *symbols* of the signature Σ.

An *expression* of Σ is any string of elements of $\mathbf{Sm}_\Sigma$. An *atomic* formula is a formula containing no logical connectives or quantifiers. Similarly to the propositional case, we can now give an interpretation of the complexity of formulae.

Definition 5.5. By the degree $d(A)$ of the formula A we understand the number of occurrences of logical connectives and quantifiers in A, with every atomic formula assigned the degree 0.

Definition 5.6. For every $A \in \mathbf{F}_\Sigma$, $x \in \Sigma_v$, and $a \in \Sigma_{ip}$ we define the formula $A^{x/a}$ as follows:

1. If A is atomic, then $A^{x/a}$ obtains by substituting a for every occurrence of x in A.
2. $(A \supset B)^{x/a} = A^{x/a} \supset B^{x/a}$.
3. $(\neg A)^{x/a} = \neg A^{x/a}$.
4. $(\forall x A)^{x/a} = \forall x A^{x/a}$.
5. $(\forall y A)^{x/a} = \forall y A^{x/a}$.

A *closed* formula, or a *sentence*, is a formula A such that, for every $a \in \Sigma_{ip}$ and every $x \in \Sigma_v$, $A^{x/a} = A$.

The notion of substitution may be easier to understand with the aid of the notion of free and bound occurrences of variables. The *scope* of an occurrence of a quantified variable, *i.e.* a variable immediately preceded by a quantifier, is the smallest formula following that occurrence.

Example 5.7. $(\forall x Px) \supset (\forall x(Qxy \supset Ry))$. Here we identify two occurrences of the universal quantifier...

The variable x has a *bound* occurrence in the formula A if it either falls within the scope of an occurrence of the quantifier in A, or else is immediately preceded by a quantifier. The variable x has a *free* occurrence if it is not bound. Then we regard the formula $A^{x/a}$ as a result of substituting a for every free occurrence of x. Equivalently, we can say that x has a free occurrence in A if $A^{x/a}$ is not the same formula as $\neq A$.

Example 5.8. If A is the formula $\forall x Px \supset \forall y Qxy$, then $A^{x/a} = \forall x Px \supset \forall y Qay$.

Definition 5.9. '$\exists x A^{x/a}$' $\iff$ '$\neg\forall x \neg A^{x/a}$'.

Remark. Metatheorems on deducibility and the deduction theorem are proved for the Hilbert-type axiomatisation of $\mathsf{T_p}$ in exactly the same way they are proved for $\mathsf{H_s}$.

[Qui51, Bos97, Men64]

Chapter 6

Predicate calculus: Semantics

6.1 Models and satisfiability

We shall now introduce several key notions of the predicate calculus. We start from afar by introducing a very general notion of algebraic system. Intuitively, an algebraic system is a set of objects, the elements of which we use in interpreting predicate parameters or functional parameters.

Definition 6.1. An *algebraic system* is a triple $\mathfrak{M} = \langle M, \Omega_F, \Omega_P \rangle$, where M is a non-empty set of individuals, Ω_F is a set of operations on M, and Ω_P is a set of predicates on M. If $\Omega_P = \varnothing$, then $\mathfrak{M}$ is called an *algebra*. If $\Omega_F = \varnothing$, then $\mathfrak{M}$ is called a *model*.

Definition 6.2. The set M of the system $\mathfrak{M} = \langle M, \Omega_F, \Omega_P \rangle$ is the *domain* of $\mathfrak{M}$. Ω_F is the *operator domain*. Ω_P is the *predicate domain*. The cardinality of $\mathfrak{M}$ is the number $|M|$.

Suppose, however, that we have to make claims about the elements of M, Ω_F, and Ω_P. We shall need a language for making those claims. Thus, an alternative formulation of the notion of algebraic system, albeit a less intuitive one, highlights the role of the signature and the distinction between object-language and metalanguage. Generally speaking, the signature may be finite.

Definition 6.3. An *algebraic system* of the countable signature Σ is a pair $\mathfrak{M} = \langle M, I \rangle$, where M is a non-empty set of individuals, and I is a mapping defined on M with the following conditions:

1. For every n-ary predicate parameter $P \in \Sigma_p$, $I(P) \subseteq M^n$;
2. For every n-ary function parameter $f \in \Sigma_f$, $I(f)\colon M^n \to M$;
3. For every $c \in \Sigma_c$, $I(c) \in M$.

We may explicitly distinguish between symbols (linguistic items) and their interpretations in $\mathfrak{M}$ by writing, for each symbol s, '$\overline{s}$' to denote $I(s)$. Sometimes it may be more convenient to replace I with the list of its values and to write $\mathfrak{M} = \langle M; \overline{P_1}, \ldots; \overline{f_1}, \ldots; \overline{c_1}, \ldots \rangle$. This notation shows also the equivalence of our two definitions of algebraic system. Moreover, let the arity of $\overline{P_i}$ and $\overline{f_i}$ be denoted by $n(\overline{P_i})$ and $n(\overline{f_i})$ respectively. Then the system $\mathfrak{M}$ is said to be of the *type* $\langle n(\overline{P_1}), \ldots; n(\overline{f_1}), \ldots \rangle$.

Example 6.4. The system $\mathfrak{Z} = \langle Z; +; \leq \rangle$ has the set of integers as the domain, the binary operation of addition in the set Ω_F and the binary predicate '① is less or equal to ②' in the set Ω_P. $\mathfrak{M}$ is of the type $\langle 2; 2 \rangle$.

Remark. In accordance with our earlier remark on representing n-ary functions with $(n+1)$-ary predicates, it is possible to transform algebras (or generally, algebraic systems) into models 'representing' them.

Here we shall only be interested in models. Given a model, we can define a valuation V for sentences of Σ.

Definition 6.5. Let $\mathbf{S}_\Sigma$ be the set of all sentences of the signature Σ. Let $\mathfrak{M} = \langle M, \Omega_P \rangle$. Let $P_M \in \Omega_P$. Then V is a *first-order* valuation of $\mathbf{S}_\Sigma$ in the model $\mathfrak{M}$ if the following holds:

1. $V(Pa_1 \cdots a_n) = \begin{cases} \mathbf{1} & \text{if } \langle a_1, \ldots, a_n \rangle \in \overline{P} \\ \mathbf{0} & \text{otherwise.} \end{cases}$
2. (a) $V(\neg A) = \begin{cases} \mathbf{1} & \text{if } V(A) = \mathbf{0} \\ \mathbf{0} & \text{if } V(A) = \mathbf{1} \end{cases}$

Table 6.1: Signed tableau processing rules for quantifiers

(b) $V(A \supset B) = \begin{cases} \mathbf{1} & \text{if } V(A) = \mathbf{0} \text{ or } V(B) = \mathbf{1} \\ \mathbf{0} & \text{if } V(A) = \mathbf{1} \text{ and } V(B) = \mathbf{0}. \end{cases}$

3. $V(\forall xA) = \begin{cases} \mathbf{1} & \text{if } V(A^{x/a}) = \mathbf{1} \text{ for all } a \in M \\ \mathbf{0} & \text{if } V(A^{x/a}) = \mathbf{0} \text{ for at least one } a \in M. \end{cases}$

4. $V(\exists xA) = \begin{cases} \mathbf{1} & \text{if } V(A^{x/a}) = \mathbf{1} \text{ for at least one } a \in M \\ \mathbf{0} & \text{if } V(A^{x/a}) = \mathbf{0} \text{ for all } a \in M. \end{cases}$

We can now define satisfiability and validity.

Definition 6.6. Let $A \in \mathbf{F}_\Sigma$. Then A is *satisfied*, or *true*, in the model $\mathfrak{M}$ if there is a valuation V of $\mathfrak{M}$ such that $V(A) = \mathbf{1}$. (Sometimes it is convenient to indicate this fact as $\mathfrak{M} \models A$.)

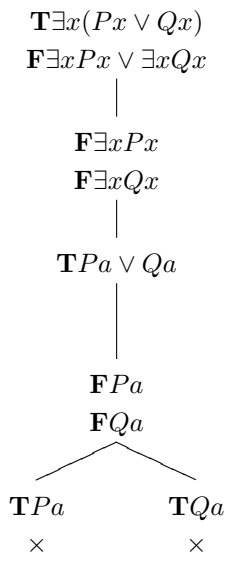
Definition 6.7. Let $A \in \mathbf{F}_\Sigma$. Then A is *valid* in the model $\mathfrak{M}$ if for all valuations V of $\mathfrak{M}$, $V(A) = \mathbf{1}$.

Definition 6.8. Let $A \in \mathbf{F}_\Sigma$. Then A is *satisfiable* if there is a model $\mathfrak{M}$ and a valuation V of $\mathfrak{M}$ such that $V(A) = \mathbf{1}$.

6.2 Rules for semantic tableaux

Tableaux have the same application in predicate calculus as they did in sentence calculus. To the earlier rules we add the rules for developing the quantifiers.

Example 6.9. Let us find a proof for $\exists x(Px \vee Qx) \vdash \exists xPx \vee \exists xQx$:



To justify the tableau method, we shall first establish the soundness of tableau proofs.

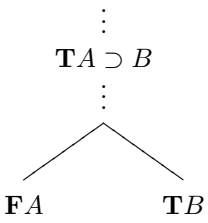
Definition 6.10. Let $A \in \mathbf{F}_\Sigma$ and $\mathfrak{M} = \langle M, I\rangle$. Then the C-sentence $A^c = A^{a_1/c_1 \cdots a_n/c_n}$ is a formula obtained from A by replacing each occurrence of individual parameters with individual constants $c_1, \ldots, c_n$ such that $\overline{c_1}, \ldots, \overline{c_n} \in M$.

The notions of satisfiability and simultaneous satisfiability naturally carry over to the case of A^c. Let us say further that a tableau $\mathscr{T}$ is simultaneously satisfiable if at least one path of it is simultaneously satisfiable.

Proposition 6.11. *Let $\mathscr{T}$ and $\mathscr{T}'$ be tableaux such that $\mathscr{T}'$ is an immediate extension of $\mathscr{T}$. Then, if $\mathscr{T}$ is simultaneously satisfiable, $\mathscr{T}'$ is also simultaneously satisfiable.*

Proof. We should consider cases for each of the tableau rules. Let us consider only selected rules.

1. Suppose $\mathscr{T}'$ is obtained from $\mathscr{T}$ by applying the rule



to the path θ of $\mathscr{T}$. Since $\mathscr{T}$ is simultaneously satisfiable, it contains a simultaneously satisfiable branch τ. If $\tau \neq \theta$, then τ is in $\mathscr{T}'$, and so $\mathscr{T}'$ is simultaneously satisfiable. Suppose $\tau = \theta$. Then θ is satisfiable. Let the model $\mathfrak{M} = \langle M, I\rangle$ simultaneously satisfy θ. Then, for a valuation V of $\mathfrak{M}$, we have that $V((A \supset B)^c) = \mathbf{1}$, where every $c_i \in M$. Thus $V((\neg A)^c) = \mathbf{1}$ or $V(B^c) = \mathbf{1}$. Clearly, then, the valuation V simultaneously satisfies either the pair $\langle \theta, \neg A\rangle$, or the pair $\langle \theta, B\rangle$. It follows that $\mathscr{T}'$ is simultaneously satisfiable, as at least one of its paths is simultaneously satisfiable.

2. Suppose $\mathscr{T}'$ is obtained from $\mathscr{T}$ by applying the rule

to the path θ of $\mathscr{T}$. Since $\mathscr{T}$ is simultaneously satisfiable, it contains a simultaneously satisfiable branch τ. If $\tau \neq \theta$, then τ is in $\mathscr{T}'$, and so $\mathscr{T}'$ is simultaneously satisfiable. Suppose $\tau = \theta$. Then θ is satisfiable. Let the model $\mathfrak{M} = \langle M, I\rangle$ simultaneously satisfy θ. Then, for a valuation V of $\mathfrak{M}$, we have that $V(\forall x(A)^c) = V((\forall x A)^c) = \mathbf{1}$, where every $c_i \in M$. Hence, in particular, $\mathfrak{M}$ satisfies $A^{x/a}$. Thus $\mathscr{T}'$ is simultaneously satisfiable. □

Proposition 6.12 (Soundness for tableaux)**.** *Let $\Gamma = \{A_1, \ldots, A_n\}$ be a set of sentences with parameters. If there is a closed finite tableau starting with $A_1, \ldots, A_n$, then Γ is not simultaneously satisfiable.*

Proof. Let $\mathscr{T}$ be a tableau starting with $A_1, \ldots, A_n$. Then there is a sequence of tableaux $\mathscr{T}_1, \mathscr{T}_2, \ldots, \mathscr{T}_n$ such that $\mathscr{T}_1$ has the sole branch $\langle A_1, \ldots, A_n\rangle$, $\mathscr{T}_n = \mathscr{T}$, and for each $0 < i < n$, $\mathscr{T}_{i+1}$ is an immediate extension of $\mathscr{T}_i$. Suppose, for *reductio*, that $\{A_1, \ldots, A_n\}$ is simultaneously satisfiable. Then $\mathscr{T}_1$ is also simultaneously satisfiable. Using induction on i and Proposition 6.11 we derive that, for each $0 < i < n + 1$, $\mathscr{T}_i$ is satisfiable. But `this is impossible, as` $\mathscr{T}_n$ `is closed by assumption`. Hence $\{A_1, \ldots, A_n\}$ is not simultaneously satisfiable. why? ... □

6.3 Logical equivalences

Before we show completeness of our method, let us record some useful equivalences of first-order calculus.

Definition 6.13. Let $A \in \mathbf{F}_\Sigma$ and let $x_1, \ldots, x_n$ be the variables occurring freely in A and $a_1, \ldots, a_n$ be the parameters not occurring in A. Let $A^a = A^{x_1/a_1 \cdots x_n/a_n}$. Then A is *satisfiable* if and only if A^a is satisfiable. And A is *logically valid* if and only if A^a is logically valid.

We can now register some properties of satisfiability and validity.

Proposition 6.14. *Let $A \in \mathbf{F}_\Sigma$. Then A is logically valid if and only if $\neg A$ is not satisfiable. And A is logically satisfiable if and only if $\neg A$ is not not logically valid.*

Proof. Straightforward from definitions. □

Proposition 6.15. *Let $A \in \mathbf{F}_\Sigma$. Then A is logically valid if and only if $\forall A$ is logically valid. A is satisfiable if and only if $\exists x A$ is satisfiable.*

Proof. Exercise. □

Definition 6.16. Let $A, B \in \mathbf{F}_\Sigma$. Then A is *logically equivalent* to B ($A \simeq B$) if and only if $A \leftrightarrow B$ is logically valid.

Proposition 6.17. *Let $A, B \in \mathbf{F}_\Sigma$ such that A and B contain a free occurrence of x. If $A \simeq B$, then $\forall x A \simeq \forall x B$ and $\exists x A \simeq \exists x B$.*

Proof. Let the formula A contain the free occurrences of the variables $x, x_1, \ldots, x_n, y_1, \ldots, y_k$, and let B contain the free occurrences of the variables $x, x_1, \ldots, x_n, z_1, \ldots, z_l$, so that the common free variables are $x, x_1, \ldots, x_n$. Since $A \simeq B$, on any arbitrary model $\mathfrak{M} = \langle M; \cdots \rangle$ the formula A will be logically valid only for those arrays $\langle \overline{x_1}, \ldots, \overline{x_n}, \overline{y_1}, \ldots, \overline{y_k}, \overline{z_1}, \ldots, \overline{z_l} \rangle$, for which B is also logically valid. But this means precisely that $\forall x A \simeq \forall x B$.

The case for the existential quantifier is proven analogously. □

Let us now formulate important logical equivalences for $\mathsf{T_p}$. We shall split them into four groups.

Proposition 6.18. *Swapping quantifiers:*

1. $\forall x \forall y P x y \simeq \forall y \forall x P x y$.
2. $\exists x \exists y P x y \simeq \exists y \exists x P x y$.

Proof. To show the first equivalence, we notice that $\forall x \forall y P x y$ is true just in case $P x y$ is logically valid. The second equivalence is left as an exercise. □

Remark. Another way of putting this claim is as follows:

1. $\forall x A \simeq A$;
2. $\exists x A \simeq A$,

assuming in both cases that x does not occur freely in A.

Proposition 6.19. *Linking existential and universal quantifiers:*

1. $\neg \forall x A \simeq \exists x \neg A$;
2. $\neg \exists x A \simeq \forall x \neg A$.

Proof. To prove 1, we notice that if the formula $\neg \forall x A$ is true, then $\forall x A$ is false. Thus the formula A is not logically valid. But then $\neg A$ is satisfiable. That is, for some $\overline{x}$ it is true. Hence, $\exists x \neg A$ is true. The other direction is proved similarly.

To prove 2, we notice that $\neg \exists x A$ is true just in case $\exists x A$ is false. This means that A is not satisfiable. And this is so just in case $\neg A$ is logically valid. And $\neg A$ is logically valid just in case $\forall x \neg A$ is logically valid. □

Remark. In the finite case, there is an analogy here with de Morgan's laws. The proved proposition may be regarded as a generalisation into the infinite case.

Proposition 6.20. *Let $A \in \mathbf{F}_\Sigma$ such that x does not occur freely in it. The distribution rules for quantifiers are as follows:*

1. $\forall x (A \wedge P x) \simeq A \wedge \forall x P x$;

2. $\forall x(A \vee Px) \simeq A \vee \forall xPx$;

3. $\exists x(A \wedge Px) \simeq A \wedge \exists xPx$;

4. $\exists x(A \vee Px) \simeq A \vee \exists xPx$;

5. $\exists x(A \supset Px) \simeq A \supset \exists xPx$;

6. $\exists x(Px \supset A) \simeq \forall xPx \supset A$;

7. $\forall x(A \supset Px) \simeq A \supset \forall xPx$;

8. $\forall x(Px \supset A) \simeq \exists xPx \supset A$.

The same claims hold if we replace the formula Px by an arbitrary formula B.

Proof. All these claims may be verified by examining the relevant tableaux. □

Proposition 6.21. *Let $A \in \mathbf{F}_\Sigma$. The rules for renaming bound variables (or rules of relettering) are as follows:*

1. $\forall xA \simeq \forall yA^{y/a}$;

2. $\exists xA \simeq \exists yA^{y/a}$.

Proof. By examining the relevant tableaux. □

6.4 Prenex normal forms

We shall now prove that any formula can be transformed into *prenex* normal form. We shall need several auxiliary notions and a lemma.

Definition 6.22. The notion of *immediate subformulae* is determined according to the following rules:

1. A and B are immediate subformulae of $A \wedge B$, $A \vee B$, $A \supset B$, whilst A is an immediate subformula of $\neg A$.

2. For any parameter a, variable x, and formula A, $A^{x/a}$ is an immediate subformula of $\forall xA$ and of $\exists xA$.

Definition 6.23. *Subformulae* are determined by the following rules:

1. If A is an immediate subformula of B, or $A = B$, then A is a subformula of B;

2. If A is a subformula of B, and B is a subformula of C, then A is a subformula of C.

Proposition 6.24. *Let $A \in \mathbf{F}_\Sigma$. If we replace its subformula B by a subformula C, such that $B \simeq C$, then the resulting formula A' will be such that $A \simeq A'$.*

Proof. Omitted. Hint: we should induce on the complexity of A. □

Definition 6.25. Let $A \in \mathbf{F}_\Sigma$. Then A is said to be in *prenex normal form* if A has the form $Q_1x_1 \cdots Q_nx_nB$, where each Q_i is a quantifier $\forall$ or $\exists$, $x_i \neq x_j$ if $i \neq j$, and B contains no quantifiers.

Proposition 6.26. *For each $A \in \mathbf{F}_\Sigma$ there is $B \in \mathbf{F}_\Sigma$ such that $A \simeq B$ and B is in prenex normal form.*

Proof. To facilitate understanding, let us first state the proof informally. We shall use the logical equivalences for $\mathsf{T_p}$ established earlier. To begin with, relying on de Morgan laws, we eliminate all symbols for material conditionals and bi-conditionals. To the resulting formula we shall apply two types of transformation. In the first transformation, we find a subformula A' of A having the form of either $C \wedge \forall xB$, or $C \vee \forall xB$, or $C \wedge \exists xB$, or $C \vee \exists xB$. For instance, let $A' = C \wedge \forall xB$ (other cases are done analogously). If C has a free occurrence of x, then we replace x by some z not occurring in A. If not, then replace A' by $\forall x(C \wedge B)$. We repeat this procedure the required number of times. In the second type of transformation, we replace the subformulae having the form $\neg\forall xB$ or $\neg\exists xB$ by $\exists x\neg B$ or $\forall x\neg B$ respectively. In this way we are able to transform A into $Q_1x_1 \cdots Q_nx_nB$ with B containing no quantifiers.

The same procedure allows a formal presentation along the following lines. Let $A \in \mathbf{F}_\Sigma$. Let $\lambda(A) \in \{0, 1, 2, \ldots\}$ be the number of occurrences of the quantifiers in A. We shall use induction on n and prove

that for a formula A with $\lambda(A) \leq n$ there is $B \in \mathbf{F}_\Sigma$ such that B is in prenex normal form, and $A \simeq B$, $\lambda(A) = \lambda(B)$, and the number of the free occurrences of variables in B is equal to the number of the free occurrences of variables in A.

Thus let $n = 0$. In this case, we may let B to be just the formula A. Now let $n > 0$. Suppose $\lambda(A) \leq n$. The quantifier-free case is trivial. Then let A contain quantifiers. If A has the form $\neg C$ and $\lambda(A) > 0$, then $\lambda(C) = \lambda(A) > 0$. By induction hypothesis there is a formula of the form QxD which is a prenex normal form for C and where $\lambda(D) = \lambda(A)$ and where D may contain quantifiers. Introduce the notation $\forall^{-1} = \exists$ and $\exists^{-1} = \forall$. We have that $A \simeq Q^{-1}x\neg D$. Now, since $\lambda(\neg D) = \lambda(QxD) = \lambda(QxA) - 1 \leq n - 1$, there is a formula $B \simeq \neg D$ which is in prenex normal form and is such that $\lambda(B) = \lambda(\neg D)$. By our equivalences above, $Q^{-1}xB \simeq A$ and $Q^{-1}xB$ satisfies the properties of the prenex normal form for A.

The proof w.r.t. the number of free variables is left as an exercise. □

Example 6.27. Let us bring the formula $\neg\exists y\neg\exists u((\exists xPxyz \supset \forall xRxy) \wedge \neg\forall zPzuz)$ to a prenex normal form:

$$\begin{aligned}
&\neg\exists y\neg\exists u((\exists xPxyz \supset \forall xRxy) \wedge \neg\forall zPzuz) \simeq\\
&\forall y\neg\neg\exists u((\neg\exists xPxyz \vee \forall xRxy) \wedge \exists z\neg Pzuz) \simeq\\
&\forall y\exists u((\forall x\neg Pxyz \vee \forall tRty) \wedge \exists v\neg Pvuv) \simeq\\
&\forall y\exists u(\forall t(\forall x\neg Pxyz \vee Rty) \wedge \exists v\neg Pvuv) \simeq\\
&\forall y\exists u(\forall t\forall x(\neg Pxyz \vee Rty) \wedge \exists v\neg Pvuv) \simeq\\
&\forall y\exists u\forall t\forall x((\neg Pxyz \vee Rty) \wedge \exists v\neg Pvuv) \simeq\\
&\forall y\exists u\forall t\forall x\exists v((\neg Pxyz \vee Rty) \wedge \neg Pvuv).
\end{aligned}$$

6.5 Skolem forms

There is a common mathematical practice of picking elements depending on the prior choice of some other elements. For instance, if we have shown that for each x there is y such that $\phi(x, y)$, then it is natural to introduce a function f^1 picking y for each x. We will then replace $\phi(x, y)$ with $\phi(x, f(x))$. Such a technique calls for the employment of a special device.

Discuss tableau rules

Consider a prenex formula $A \in \mathbf{F}_\Sigma$. It contains pairwise distinct variables $x_1, \ldots, x_n$, quantifiers $Q_1, \ldots, Q_n$, and a quantifier-free formula $B \in \mathbf{F}_\Sigma$. Then we can first identify the indices of the existential quantifiers: $\{j_1, \ldots, j_i, \ldots, j_p \mid Q_{j_i} = \exists, 1 \leq i \leq n\}$. Given such a set, we can now expand the signature Σ into Σ^A_{Sk} by adding p new symbols for parameters or functions. Those will be symbols for *Skolem functions* (sometimes also called 'Herbrand functions') associated with A. We may also compute the arity of the particular symbol f_h. For $1 \leq h \leq p$, its arity will equal the number of times the universal quantifier occurs to the left of Q_{j_h} in the prefix of A. That would be exactly the number $j_h - h$. We also note that constants may be regarded as 0-ary function symbols. Therefore, for f_h to be a constant, we require that $j_h = h$, or equivalently, that the first h quantifications be existential.

Example 6.28. Let the prefix of A be:

$$\forall x_0\forall x_1\forall x_2\exists x_3\exists x_4\forall x_5\exists x_6\forall x_7\forall x_8\exists x_9\forall x_{10}.$$

Then we expand Σ into Σ^A_{Sk} by adding four new function symbols f_1, f_2, f_3, f_4, whose respective arities will be $3, 3, 4, 6$.

We can now build the *Skolem form* A_{Sk} of the formula A. Obviously it will be a prenex formula containing only universal quantifiers. Let u_h be a term of Σ^A_{Sk} consisting of the function symbol f_h followed by $j_h - h$ universally quantified variables such that they occur to the left of the variable x_{j_h} in the prefix of the formula A. In general, u_h will take the form:

$$f_h x_1 x_2 \cdots x_{j_1-1} x_{j_1+1} \cdots x_{j_2-1} x_{j_2+1} \cdots x_{j_{h-1}-1} x_{j_{h-1}+1} \cdots x_{j_h-1}.$$

Then, for each $1 \leq h \leq n$ we replace each occurrence of x_{j_h} in B by the term u_h. And in front of this formula we put the prefix of A from which each occurrence of the existential quantifier has been deleted.

Example 6.29. Suppose the signature Σ contains predicate parameters P^1 and R^2. Consider $A \in \mathbf{F}_\Sigma$:

$$\exists x_0\exists x_1\forall x_2\exists x_3\forall x_4\forall x_5\exists x_6((Rx_0x_2 \wedge Px_5) \supset (Rx_6x_2 \vee (Rx_1x_5 \wedge Rx_4x_3))).$$

The signature Σ^A_{Sk} will then contain four new symbols: two constants f_1 and f_2, a unary function parameter f_3 and a ternary function parameter f_4. The formula A_{Sk} will take the form:

$$\forall x_2 \forall x_4 \forall x_5((Rf_1x_2 \wedge Px_5) \supset (Rf_4(x_2, x_4, x_5)x_2 \vee (Rf_2x_5 \wedge Rx_4f_3(x_2)))).$$

We must keep in mind that A_{Sk} belongs to a richer signature than A. Hence it is wrong to say that A is equivalent to its Skolem form. What is true, however, is that when A is considered as a formula belonging to Σ^A_{Sk}, then it will be semantically entailed by its Skolem form.

Example 6.30. Let us illustrate the claim of entailment. Let A be the formula $\forall x_0 \exists x_1 Rx_0x_1$. We obtain its Skolem form by adding the function parameter g to its signature Σ. The formula A_{Sk} will be $\forall x_0 Rx_0g(x_o)$. Let $\mathfrak{M} = \langle M; R, g\rangle$ be the algebraic system such that $\mathfrak{M} \models A_{\mathrm{Sk}}$. Thus we have that for every $a \in M$ we have that $\langle a, g(a)\rangle \in R$. But this means that $\mathfrak{M} \models A$. Therefore, $A_{\mathrm{Sk}} \vDash A$.

Skolem functions and Skolem forms have an interesting application in proof theory, as we shall see later in our discussion of Gentzen systems. A special case of Skolem functions is also used in one of the proofs of first-order completeness. But their most vital role belongs in model theory. One important fact there is that for a closed formula to be satisfiable it is necessary and sufficient that its Skolem form be satisfiable. Most of these applications will fall outside the scope of our concerns. Here we shall only formulate a basic property of Skolem forms:

Proposition 6.31. *Let $y_1, \ldots, y_n$ be pairwise distinct variables and let $A \in \mathbf{F}_\Sigma$ be a prenex formula with free occurrences of $y_1, \ldots, y_n$. Then the formula $A_{\mathrm{Sk}} \supset A$ of the signature Σ^A_{Sk} is valid.*

Proof. To be supplied. □

6.6 Completeness for tableaux

We now resume our enquiry into the completeness of the tableau method. Let M be a domain of individuals and let Γ be a set of signed C-sentences (which, we recall, should be regarded as closed C-formulae) associated with it. We say that Γ is *closed* if it contains a conjugate pair of C-sentences. Of particular interest to us will be those sets of C-sentences where, so to speak, all the sentences have been processed according to the tableau rules. Thus:

Definition 6.32. Let Γ be a set of signed C-sentences. Then Γ is *well-developed on M* if Γ obeys the rules for semantic tableaux, so that the following conditions hold:

1. If $\mathbf{T}\neg A \in \Gamma$, then $\mathbf{F}A \in \Gamma$, and if $\mathbf{F}\neg A \in \Gamma$, then $\mathbf{T}A \in \Gamma$;
2. If $\mathbf{T}A \wedge B \in \Gamma$, then $\mathbf{T}A \in \Gamma$ and $\mathbf{T}B \in \Gamma$, and if $\mathbf{F}A \wedge B \in \Gamma$, then either $\mathbf{F}A \in \Gamma$ or $\mathbf{F}B \in \Gamma$;
3. If $\mathbf{T}A \vee B \in \Gamma$, then either $\mathbf{T}A \in \Gamma$ or $\mathbf{T}B \in \Gamma$, and if $\mathbf{F}A \wedge B \in \Gamma$, then $\mathbf{F}A \in \Gamma$ and $\mathbf{F}B \in \Gamma$;
4. If $\mathbf{T}A \supset B \in \Gamma$, then either $\mathbf{F}A \in \Gamma$ or $\mathbf{T}B \in \Gamma$, and if $\mathbf{F}A \wedge B \in \Gamma$, then $\mathbf{T}A \in \Gamma$ and $\mathbf{F}B \in \Gamma$;
5. (The condition for the bi-conditional is left as an exercise.)
6. If $\mathbf{T}\exists xA \in \Gamma$, then $\mathbf{T}A^{x/a} \in \Gamma$ for at least one $a \in M$, and if $\mathbf{F}\exists xA \in \Gamma$, then $\mathbf{F}A^{x/a} \in \Gamma$ for all $a \in M$;
7. If $\mathbf{T}\forall xA \in \Gamma$, then $\mathbf{T}A^{x/a} \in \Gamma$ for all $a \in M$, and if $\mathbf{F}\forall xA \in \Gamma$, then $\mathbf{F}A^{x/a} \in \Gamma$ for at least one $a \in M$.

(Well-developed sets are called 'Hintikka sets' in [Smu68].)

To prove completeness, we shall first establish several lemmas.

Proposition 6.33 (Hintikka's Lemma)**.** *Let Γ be a set of signed C-sentences. If Γ is well-developed on M and open, then Γ is simultaneously satisfiable.*

Proof. Let us assume that Γ is well-developed on M and open. Consider a model $\mathfrak{M} = \langle M, I\rangle$ of the signature Σ, such that for any predicate parameter $P^n \in \Sigma_p$, $\overline{P^n} = \{\langle a_1, \ldots, a_n\rangle \in M^n \mid \mathbf{T}Pa_1 \cdots a_n \in \Gamma\}$. We now claim that for any C-sentence A the following holds:

i. If $\mathbf{T}A \in \Gamma$, then $V(A) = \mathbf{1}$;

ii. If $\mathbf{F}A \in \Gamma$, then $V(A) = \mathbf{0}$.

The proof should be by induction on the complexity of A. As the basic step, we consider the formula A such that $d(A) = 0$. The inductive hypothesis will be that if $d(A) > 0$, then for any formula B such that $d(B) < d(A)$ our claims hold. We shall have to consider all the cases corresponding to each of the tableau rules. Let us consider here some of them, and the rest will be left as an exercise.

1. Let $d(A) = 0$. Then A will have the form $Pa_1 \cdots a_n$.
   i. If $\mathbf{T}Pa_1 \cdots a_n \in \Gamma$, then we already we have that $\langle a_1, \ldots, a_n \rangle \in \overline{P^n}$, and so it follows that $V(Pa_1 \cdots a_n) = \mathbf{1}$.
   ii. If $\mathbf{F}Pa_1 \cdots a_n \in \Gamma$, then we already we have that $\langle a_1, \ldots, a_n \rangle \notin \overline{P^n}$, and so it follows that $V(Pa_1 \cdots a_n) = \mathbf{0}$.
2. Suppose $d(A) > 0$ and A has the form $\neg B$. Then $d(A) > d(B)$, and so the inductive hypothesis holds for B.
3. Suppose $d(A) > 0$ and A has the form $B \wedge C$. Then $d(A) > d(B)$ and $d(A) > d(C)$, and so the inductive hypothesis holds for B and C.
   i. If $\mathbf{T}B \wedge C \in \Gamma$, then, since Γ is well-developed, we already we have that $\mathbf{T}B, \mathbf{T}C \in \Gamma$, and so it follows that $V(B) = V(C) = \mathbf{1}$. Hence $V(B \wedge C) = \mathbf{1}$.
   ii. If $\mathbf{F}B \wedge C \in \Gamma$, then we already we have that $\mathbf{F}B \in \Gamma$ or $\mathbf{F}C \in \Gamma$, and so it follows that either $V(B) = \mathbf{0}$ or $V(C) = \mathbf{0}$. Hence $V(B \wedge C) = \mathbf{0}$.
4. Suppose $d(A) > 0$ and A has the form $\exists xB$. Then for all $a \in M$ we have $d(B^{x/a}) < d(A)$, and so the inductive hypothesis holds for $B^{x/a}$.
5. Suppose $d(A) > 0$ and A has the form $\forall xB$. Then for all $a \in M$ we have $d(B^{x/a}) < d(A)$, and so the inductive hypothesis holds for $B^{x/a}$. □

For the following lemma we let $N = \{a_1, \ldots, a_n, \ldots\}$ be a set of parameters.

Proposition 6.34. *Let $\mathscr{T}_0$ be a finite tableau. By applying tableau rules it is possible to extend $\mathscr{T}_0$ to a possibly infinite tableau $\mathscr{T}$ such that every closed path of $\mathscr{T}$ is finite and every open path of it is well-developed on N.*

Proof. We construct an array of finite extensions of $\mathscr{T}_0$: $\mathscr{T}_1, \mathscr{T}_2, \ldots, \mathscr{T}_i, \ldots,$. If for some i the tableau $\mathscr{T}_i$ is closed, then the procedure halts and we let $\mathscr{T} = \mathscr{T}_i$. But in any case we may set $\mathscr{T} = \mathscr{T}_\infty = \bigcup_{i=0}^{\infty} \mathscr{T}_i$.

We proceed as follows. Let us call the node $X \in \mathscr{T}_0$ 'semi-universal', if it has the form $\mathbf{T}\forall xA$ of $\mathbf{F}\exists xA$. Suppose we have constructed $\mathscr{T}_i$. Then for each semi-universal node $X \in \mathscr{T}_i$ and each $n \leq i$, we apply the relevant tableau rule to extend each open path of $\mathscr{T}_i$ containing X by $\mathbf{T}A^{x/a_n}$ or $\mathbf{F}A^{x/a_n}$. Let $\mathscr{T}_{i+1}$ be the tableau obtained as a result. Then for each node X of $\mathscr{T}_{i+1}$ which is not semi-universal we extend each open path containing X by applying the relevant tableau rule. And then again repeat the procedure for the finite tableau $\mathscr{T}_{i+2}$. Therefore, a closed path is never extended, so that all closed paths of $\mathscr{T}_\infty$ are finite. It is also clear from our construction that each open path of $\mathscr{T}_\infty$ is well-developed on N. Therefore, $\mathscr{T}_\infty$ has the desired properties. □

Finally, let us prove another lemma related to the properties of trees generally. Recall that in a finitely generated tree each node has finitely many successors (*i.e.* 'immediate' successors).

Proposition 6.35 (König's Lemma)**.** *Let $\mathscr{T}$ be a finitely generated tree with infinitely many nodes. Then $\mathscr{T}$ contains at least one infinite path.*

Proof. Let $\mathscr{T}_x$ be the set of all nodes $x \in \mathscr{T}$ such that x has infinitely many descendants (*i.e.* both 'immediate' and non-'immediate' successors). Then $\mathscr{T}_x$ is a sub-tree of $\mathscr{T}$. Each $x \in \mathscr{T}_x$ will have at least one successor. Let x_1 be the root of $\mathscr{T}$. By assumption, since $\mathscr{T}$ is infinite, the root will have infinitely many descendants. If all of the successors of x_1 have finitely many descendants, then x_1 would have had finitely many descendants—which is false. Hence there is at least one successor of x_1 which has infinitely many descendants. Let it be x_2. We thus construct inductively a sequence $x_1, x_2, \ldots,$. Clearly such a sequence will constitute an infinite path of $\mathscr{T}$. □

No need for the axiom of choice

We are now in a position to prove completeness. Similarly to the propositional case (Proposition 4.47), we are going to connect the semantic notion of validity with the syntactic notion of provability. Recall that the provability of A by the tableau method would mean that there is a closed tableau with $\mathbf{F}A$ at the root.

Proposition 6.36 (Completeness)**.** *Let $\Gamma = \{A_1, \ldots, A_n\}$ a set of sentences with parameters. If Γ is not simultaneously satisfiable, then there is a finite closed tableau with Γ at the root.*

Proof. By Proposition 6.34 there is a tableau $\mathscr{T}$ such that every closed path of it is finite and every open path of it is well-developed on M. If $\mathscr{T}$ is closed, then by König's Lemma $\mathscr{T}$ is finite. Suppose $\mathscr{T}$ is open. Let Δ be its open path. Then Δ is well-developed. But by Hintikka's Lemma Δ is simultaneously satisfiable in M. Since $\Gamma \subseteq \Delta$, Γ is simultaneously satisfiable in M. □

The following claims are obtained as simple corollaries of the completeness result as stated above.

Proposition 6.37 (Löwenheim)**.** *Let $\Gamma = \{A_1, \ldots, A_n\}$ a set of sentences with parameters. If Γ is simultaneously satisfiable, then it is simultaneously satisfiable in a countable domain.*

Proof. Follows from the proof of Proposition 6.36. □

Proposition 6.38. *Let $\Gamma = \{A_1, \ldots, A_n\}$ a set of sentences with parameters. Γ is not simultaneously satisfiable if and only if there exists a finite closed signed tableau starting with $\mathbf{T}A_1, \ldots, \mathbf{T}A_k$.*

Proposition 6.39. *The sentence A is logically valid if and only if there exists a finite closed signed tableau starting with $\mathbf{F}A$.*

Proposition 6.40. *The sentence B is a logical consequence of $A_1, \ldots, A_k$ if and only if there exists a finite closed signed tableau starting with $\mathbf{T}A_1, \ldots, \mathbf{T}A_k, \mathbf{F}B$.*

We shall now prove the compactness theorem for predicate calculus, a much deeper result than compactness for propositional calculus. It was originally proven by Mal'cev, of course using different techniques. We again prove it for the countable case, since the uncountable case invokes the axiom of choice.

Proposition 6.41 (Compactness)**.** *Let Γ be a countably infinite set of first-order sentences. Then Γ is simultaneously satisfiable if and only if each finite subset of Γ is simultaneously satisfiable.*

Proof. Let $\Gamma = \{A_0, A_1, \ldots, A_i, \ldots\}$. We start by letting $\mathscr{T}_0$ be the empty tableau. Suppose we have constructed $\mathscr{T}_i$. Extend $\mathscr{T}_i$ to $\mathscr{T}_i'$ by appending A_i to each open path of $\mathscr{T}_i$. Since $\{A_0, A_1, \ldots, A_i\}$ is simultaneously satisfiable, $\mathscr{T}_i'$ has at least one open path. Now extend $\mathscr{T}_i'$ to $\mathscr{T}_{i+1}$ and then to $\mathscr{T}_{i+2}$ as in the proof of Proposition 6.34. Finally we let $\mathscr{T} = \mathscr{T}_\infty = \bigcup_{i=0}^{\infty} \mathscr{T}_i$. We have that every closed path of $\mathscr{T}$ is finite, and every open path of $\mathscr{T}$ is well-developed on M. Note also that $\mathscr{T}$ is an infinite, finitely branching tree. By König's Lemma we let Γ' be an infinite path in $\mathscr{T}$. Then Γ' is well-developed on M and $\Gamma \subseteq \Gamma'$. By Hintikka's Lemma, Γ' is simultaneously satisfiable. Hence Γ is simultaneously satisfiable. □

We can now prove another well-known theorem which strengthens the result of Proposition 6.37.

Proposition 6.42 (Skolem-Löwenheim)**.** *Let Γ be a countably infinite set of first-order sentences. If Γ is simultaneously satisfiable, then Γ is simultaneously satisfiable in a countable domain.*

Proof. Follows from the proof of the compactness theorem. Notice that Γ is well-developed in the countably infinite domain M, and therefore, simultaneously satisfiable on it. □

Let us now go back to the completeness result we have obtained. It appears as though the tableau method provides a test for logical validity of sentences of the predicate calculus. Unfortunately, the test would only be partially effective. If a sentence A is logically valid, we will certainly find a finite closed tableau starting with $\mathbf{F}A$. But if A is not logically valid, we will not necessarily find a finite tableau to show this.

Example 6.43. Consider the formula:

$$A : \forall x \exists y Pxy \wedge \forall x \forall y (Pxy \supset \neg Pyx) \wedge \forall x \forall y \forall z (Pxy \supset (Pyz \supset Pxz)).$$

This formula is satisfiable on an infinite model, but is not satisfiable on any finite one. Consider $\mathfrak{M} = \langle \{0, 1, 2, \ldots\}; \overline{<} \rangle$. Clearly $\mathfrak{M} \models A$. To see that A is not satisfiable on a finite model, let $\mathfrak{M} = \langle M; \overline{P} \rangle$ such that $\mathfrak{M} \models A$. Note that $V(Paa) = \mathbf{0}$ for any $\overline{a} \in M$. Let $\overline{a_0} \in M$. Now select $\overline{a_1} \in M$ such that $V(Pa_0a_1) = \mathbf{1}$. Then select $\overline{a_2} \in M$ such that $V(Pa_1a_2) = \mathbf{1}$. By repeating the process we get a sequence $\langle a_0, a_1, a_2, \ldots \rangle$. All the members of this sequence are different: if $i < j$, then $V(P(a_ia_{i+1})) = V(P(a_{i+1}a_{i+2})) = \cdots = V(P(a_{j-1}a_j)) = \mathbf{1}$. Since $\mathfrak{M} \models \forall x \forall y \forall z (Pxy \supset (Pyz \supset Pxz))$, we have that $V(P(a_ia_j)) = \mathbf{1}$. So $a_i \neq a_j$. Therefore, M is (countably) infinite.

The same conclusion may be reached by examining the tableau for A in Figure 6.1. Its infinite open path gives rise (see the proof of Proposition 6.33) to an infinite model, such as $\mathfrak{M}$ above.

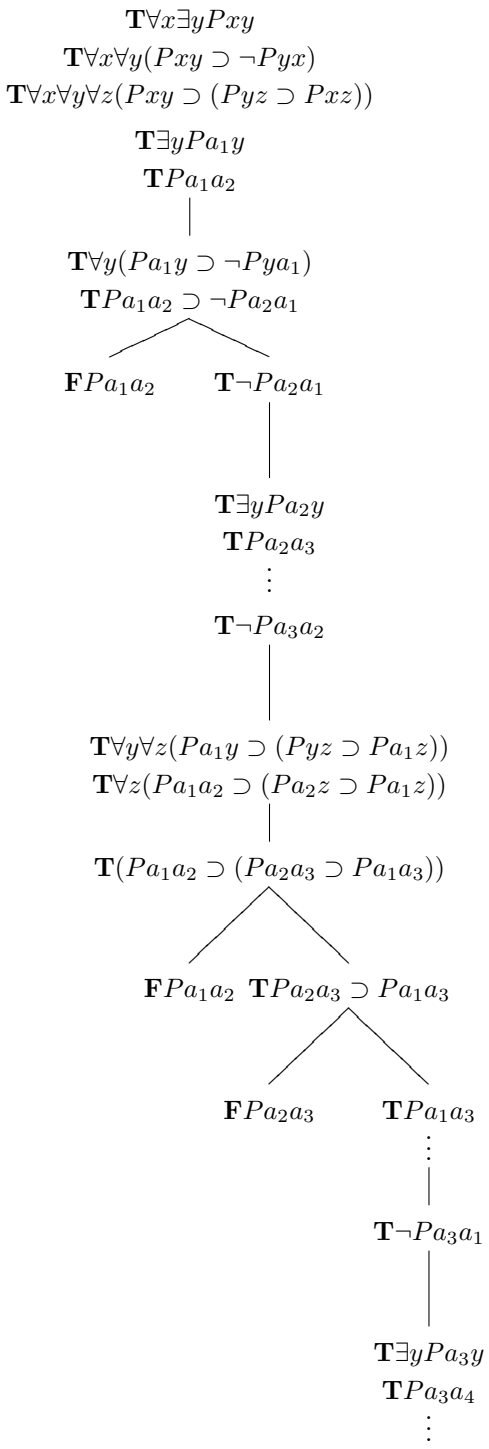


Figure 6.1: Tableau for a formula satisfiable in an infinite domain.

.

6.7 Normal models

[Bos97, Smu68]

Chapter 7

Modal logic

7.1 Modalities

Modalities (or modals) serve to qualify the truth of a statement:

Roger is $\cdots\cdots$ rich.

Instead of the dots we could put a qualifier such as 'necessarily', 'possibly', 'now', 'then', 'believed by me to be' and so forth. We could also paraphrase that statement to get:

It is $\cdots\cdots$ true that Roger is rich.

So it does not matter whether we qualify the predicate or the whole statement.

We are interested here in necessity and possibility. Many people people believe that things could be different from the way they actually are. Nadal could have won Wimbledon—if Federer had a flu, perhaps. We can isolate the statement about Nadal:

Nadal could win Wimbledon.

We go further and put the modal verb at the front:

It could be that Nadal won Wimbledon.

The possibility indicated by the modal verb 'could' will be represented by a modal operator. A further paraphrase:

Possibly Nadal won Wimbledon. (1)

Now one sense of (1) is epistemic whereby it becomes equivalent to 'For all we know, Nadal won Wimbledon.' This is *not* the meaning we attribute to (1) and synonymous sentences. We take them to mean that there are circumstances alternative to the actual circumstances—for example, the circumstance of Nadal being victorious at Wimbledon.

Necessity and possibility are represented by modal operators $\Box$ and $\Diamond$. Accordingly, to our classical clauses for well-formed formulae we add another one:

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, then so is $\neg A$;

FR3. If A and B are formulae, then so is $(A \supset B)$;

FR4. If A is a formula, then so are $\Box A$ and $\Diamond A$.

7.2 Modal square

What is the relation between necessity and possibility? Aristotle worked out the basic connections.

7.3 Semantics

Consider again our modal modifier:

Roger is $\cdots\cdots$ is rich.

We can put the modifier at the end:

Roger is rich $\cdots\cdots$. (1)

We can the modifier to refer to a particular circumstance. Hence, if Roger is rich necessarily, he is so in *every* circumstance. If he is rich possibly, he is so in *some* circumstance. In the classical logic we have already seen that circumstances are fixed by valuations. Therefore, (1) is paraphrased as:

Roger is rich under valuation i.

Necessity and possibility will accordingly take the form:

For all i, Roger is rich in i.
For some i, Roger is rich in i.

Example 7.1. Consider the formula $\Box P \supset P$. It says that if P is true under every valuation, it is true under the actual valuation. So this is an intuitively valid claim. By contrast, consider $P \supset \Box P$. Merely because P is true in the actual circumstance, it is not true in every other circumstance. So this claim is invalid.

Possible valuations are commonly known as 'possible worlds'. Modal semantics uses another, not quite so intuitive notion of *accessibility* between worlds. Let us illustrate it with following example.

Example 7.2 (Accessibility)**.** Let $\Diamond P$ be interpreted as 'It is conceivable that P' and consider the formula:

$$\Diamond P \supset \Box\Diamond P. \tag{2}$$

The formula (2) says that if P is true in some conceivable world—the one we can conceive from the actual world—then from every conceivable world there is a conceivable world in which P is true. Now from the perspective of a medieval monk it is inconceivable that the Earth is four billion years old. But it is conceivable to us. So the antecedent of (2) is true, but the consequent is false.

Let us proceed with a formal treatment of semantics.

Definition 7.3. A *frame* is a pair $\langle \mathscr{W}, \mathscr{R} \rangle$ where $\mathscr{W}$ is a set of possible worlds and $\mathscr{R}$ is the accessibility relation defined on $\mathscr{W}$.

Definition 7.4. A *propositional modal model* is a triple $\langle \mathscr{W}, \mathscr{R}, \Vdash \rangle$, where $\Vdash$ is a relation between possible worlds and sentential parameters. In case $\Gamma \Vdash P$ holds, we say that P is true on Γ, otherwise we say it is false on Γ.

Definition 7.5. Let $\langle \mathscr{W}, \mathscr{R}, \Vdash \rangle$ be a model. Then for each $\Gamma \in \mathscr{W}$:

1. $\Gamma \Vdash \neg A$ iff $\Gamma \nVdash A$
2. $\Gamma \Vdash A \supset B$ iff $\Gamma \nVdash A$ or $\Gamma \Vdash B$
3. $\Gamma \Vdash \Box A$ iff for every $\Delta \in \mathscr{W}$, if $\Gamma \mathscr{R} \Delta$, then $\Delta \Vdash A$

To illustrate the significance of the accessibility relation and the notion of truth in a modal model, we shall provide diagrams.

Example 7.6. See Figure 7.1. Here we have that, since $\Delta \Vdash P$, $\Delta \Vdash P \vee Q$. Similarly $\Omega \Vdash P \vee Q$. Now $\Gamma \Vdash \Box(P \vee Q)$. But neither $\Gamma \Vdash \Box P$, nor $\Gamma \Vdash \Box Q$. So $\Gamma \nVdash (\Box P \vee \Box Q)$. Therefore, $\Gamma \nVdash (\Box(P \vee Q) \supset (\Box P \vee \Box Q))$.

Example 7.7. See Figure 7.2. Here we have that $\Gamma \nVdash (\Diamond P \supset \Box\Diamond P)$.

We now turn to the issue of showing that some formulae are true at the worlds. The diagrams do not help as they provide *counter-examples*.

Proposition 7.8. *Let $\langle \mathscr{W}, \mathscr{R}, \Vdash \rangle$ be a model, and let $\Gamma \in \mathscr{W}$. Then, if $\Gamma \Vdash \Box(P \wedge Q)$, then $\Gamma \Vdash (\Box P \wedge \Box Q)$.*

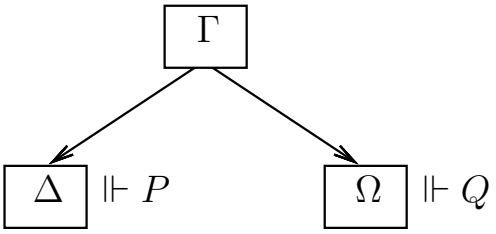


Figure 7.1: No distribution of necessity

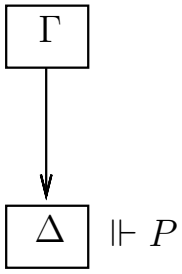


Figure 7.2: No necessitation of possibility

Proof. Working straight from definitions:

$\Gamma \Vdash \Box(P \wedge Q)$	
$\Delta \in \mathscr{W} : \Gamma \mathscr{R} \Delta$	Δ is arbitrary
$\Delta \Vdash (P \wedge Q)$	
$\Delta \Vdash P$	
$\Delta \Vdash Q$	
$\Gamma \Vdash \Box P$	since Δ is arbitrary
$\Gamma \Vdash \Box Q$	since Δ is arbitrary
$\Gamma \Vdash (\Box P \wedge \Box Q)$.	

□

Proposition 7.9. *If $\mathscr{R}$ is reflexive, then, for every $\Gamma \in \mathscr{W}$, $\Gamma \Vdash (\Box P \supset P)$.*

Proof. We prove by *reductio*:

$\Gamma \Vdash \Box P$	
$\Gamma \mathscr{R} \Gamma$	reflexivity
$\Gamma \Vdash P$	
$\Gamma \nVdash P$	For *reductio*
$\bot$.	

□

Proposition 7.10. *If $\mathscr{R}$ is symmetric, then, for every $\Gamma \in \mathscr{W}$, $\Gamma \Vdash (P \supset \Box \Diamond P)$.*

Proof. Again by *reductio*:

$\Gamma \Vdash P$	
$\Gamma \nVdash \Box\Diamond P$	For *reductio*
$\Gamma \mathscr{R} \Delta : \Delta \nVdash \Diamond P$	
$\forall w \in \mathscr{W} : \Delta \mathscr{R} w \Rightarrow w \nVdash P$	
$\Delta \mathscr{R} \Gamma$	symmetry
$\Gamma \nVdash P$	
$\bot$.	

□

Proposition 7.11. *If $\mathscr{R}$ is symmetric and transitive, then, for every $\Gamma \in \mathscr{W}$, $\Gamma \Vdash (\Diamond P \supset \Box\Diamond P)$.*

Proof. Again by *reductio*:

$\Gamma \Vdash \Diamond P$	
$\Gamma \mathscr{R} \Omega : \Omega \Vdash P$	
$\Gamma \nVdash \Box\Diamond P$	For *reductio*
$\Gamma \mathscr{R} \Delta : \Delta \nVdash \Diamond P$	
$\forall w \in \mathscr{W} : \Delta \mathscr{R} w \Rightarrow w \nVdash P$	
$\Delta \mathscr{R} \Gamma$	symmetry
$\Delta \mathscr{R} \Omega$	transitivity
$\Omega \nVdash P$	
$\bot$.	

□

7.4 Tableau rules

We modify familiar tableau rules to fit the semantics for modal logic.

Definition 7.12. A *modal prefix* is a finite sequence of positive integers. A prefixed formula is an expression of the form $\xi\, A$, where ξ is a modal prefix and A is a signed formula.

Example 7.13. '1.2.4.7' is a prefix. Moreover, if ξ is a prefix and n is a positive integer, then $\xi.n$ is also a prefix.

Intuitively the prefix designates a possible world, and $\xi\, A$ says that A is true on ξ. The prefix $\xi.n$ designates a possible world accessible from ξ. The modified tableau rules are given in Table 7.4. All the definitions and applications of the tableau method carry over from the classical case, with the following amendment:

Definition 7.14. A *conjugate* pair of formulae is a pair $\langle \xi\, \mathbf{T}A, \xi\, \mathbf{F}A \rangle$.

Our rules correspond to the basic modal system **K** in which no restrictions are imposed on relation $\mathscr{R}$. It is a very weak system in which many intuitive properties of necessity and possibility cannot be proven.

Remark. We have to exercise care in applying the rule for necessity. The rule says that we can proceed from $\xi\, \mathbf{T}\Box A$ to $\xi.n\, \mathbf{T}A$ *provided* there is a world designated by $\xi.n$—that is, there is a world accessible from ξ. The difficulty here corresponds to the situation in the predicate logic when we work with an empty domain. It can be illustrated by the tableau in Figure 7.3 where we attempt to prove the formula $\Box A \supset \Diamond A$. Accordingly, we in effect should make the proviso that the necessity-like rules can employ the prefix $\xi.n$ if it already occurs in the tableau above, thus limiting the analogy with the universal quantifier rules.

Example 7.15. Show that $\vDash (\Box P \wedge \Box Q) \supset \Box(P \wedge Q)$. The desired tableau is in Figure 7.4.

The properties of necessity and possibility may be explored either syntactically or semantically. The syntactic way means having additional tableau rules governing the operators of necessity and possibility. The semantic way means imposing restrictions on the accessibility relation in the modal model. We have seen

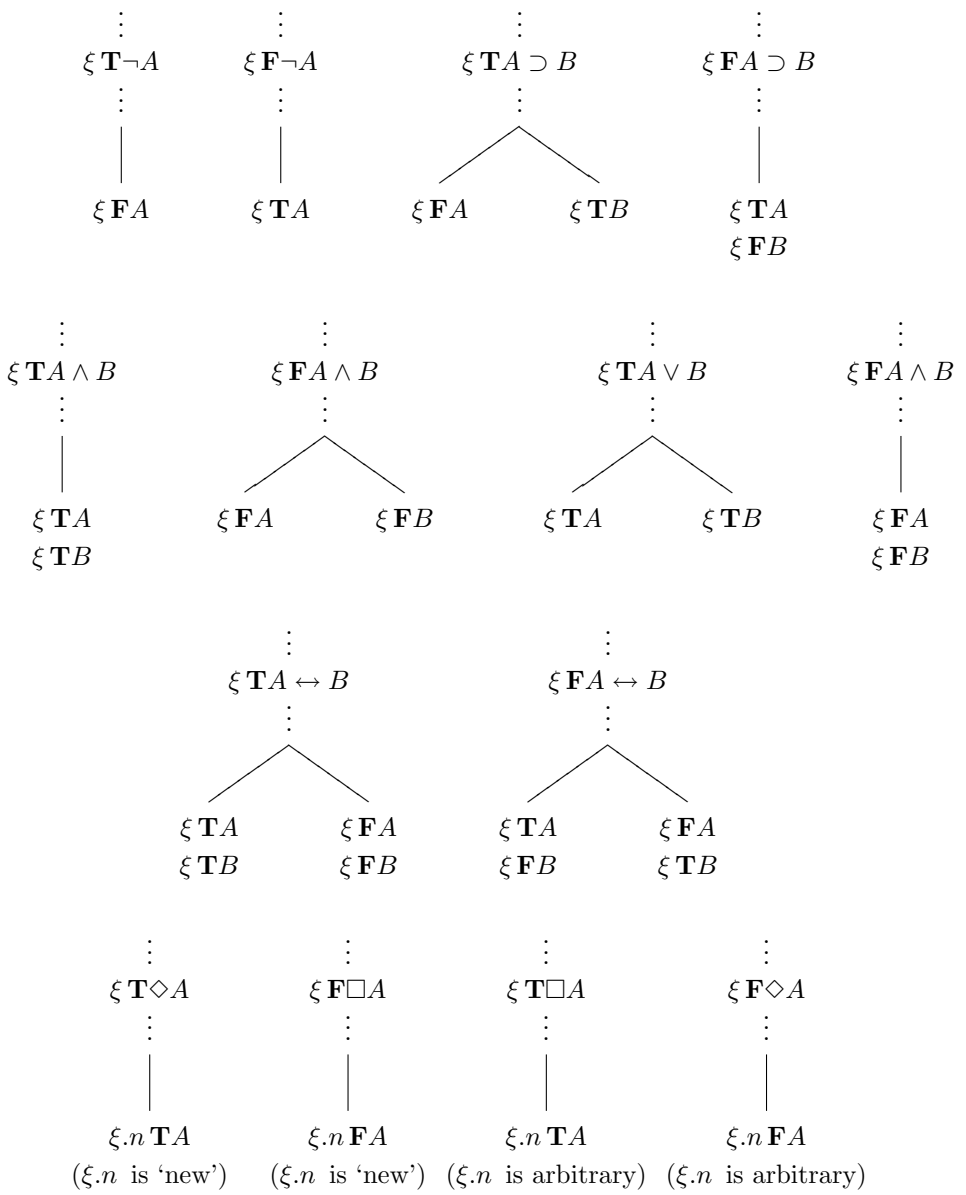


Table 7.1: Signed modal tableau processing rules

Figure 7.3: $\Box A \supset \Diamond A$ is not **K**-valid.

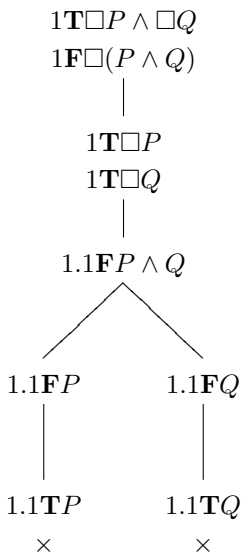


Figure 7.4: $(\Box P \wedge \Box Q) \supset \Box(P \wedge Q)$ is **K**-valid.

above in Propositions 7.9, 7.10, and 7.11 that the two ways are correlated with each other (at least in some cases). The special tableau rules are given in Table 7.4. The relevant logics obtain as follows:

D	D	serial
T	T	reflexive
K4	4	transitive
B	$B, 4$	reflexive, symmetric
S4	$T, 4$	reflexive, transitive
S5	$T, 4, 4r$	equivalence.

Example 7.16. Let us show that $\Diamond(A \wedge \Box B) \leftrightarrow (\Diamond A \wedge \Box B)$ is **S5**-valid. The relevant tableau is in Figure 7.5.

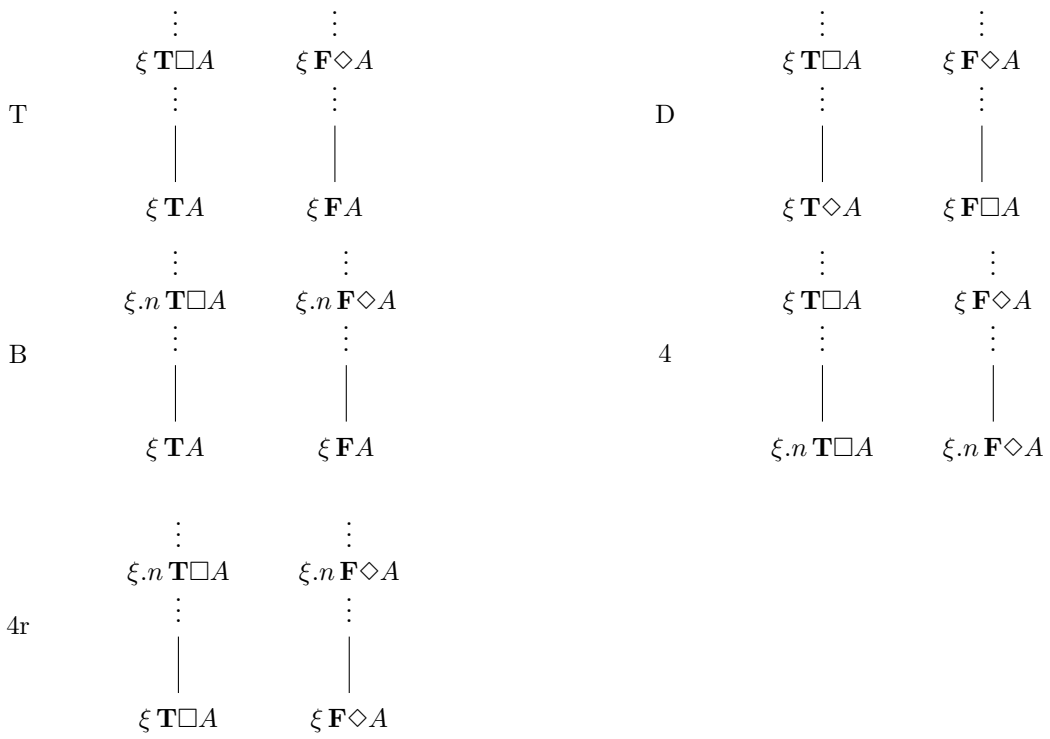


Table 7.2: Additional tableau processing rules

7.5 Digression: axioms

Let us first list the additional axioms for some particularly important logics:

K	$\Box(A \supset B) \supset (\Box A \supset \Box B)$
D	$\Box A \supset \Diamond A$
T	$\Box A \supset A$
4	$\Box A \supset \Box\Box A$
B	$A \supset \Box\Diamond A$
5	$\Diamond A \supset \Box\Diamond A$.

The relevant logics obtain as follows:

K	K
D	D
T	T
K4	4
B	T, B
S4	$T, 4$
S5	$T, 5$ or $T, 4, B$.

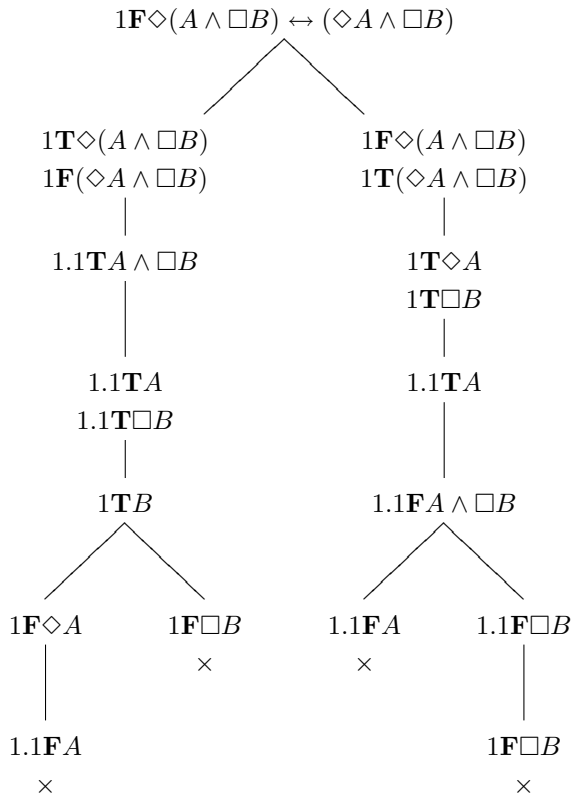


Figure 7.5: $\Diamond(A \wedge \Box B) \leftrightarrow (\Diamond A \wedge \Box B)$ is **S5**-valid.

Chapter 8

Computability

Historically, issues of computability derive from the problem of finding values of functions. A computing procedure serving such a purpose was labelled 'algorithm' (the word itself comes from the name of al-Khoresmi, a Baghdad mathematician of the ninth century). Instances of algorithms include such procedures as finding the greatest common divisor of two natural numbers, solving a system of two linear equations, and so forth. We can similarly talk of algorithms purporting to identify properties of the given relation. Thus we may look for an algorithm answering the question whether a certain natural number is prime, or whether certain two numbers are relatively prime.

8.1 The notion of an algorithm

Intuitively given instances of computing procedures indicate the following properties of the algorithm:

Unambiguous description. An algorithm must have such a description that would allow us to conduct it mechanically, without a need of attending to its mathematical (or logical, or other) content.

Determinism. Every step of the algorithm must be uniquely determined by its previous steps and the initial data.

Isolation. Computation must be conducted only by the algorithm's instructions without any interference of external processes or computing devices.

With time the increase in the complexity of computing tasks lead to a situation where no concrete procedures were found. A hypothesis was raised that perhaps *no* algorithms exist for those tasks. The theoretical impossibility of finding such algorithms cannot be decided by any number of examples. It calls for a general solution. Therefore, a verdict on this problem cannot be reached unless we refine our intuitive notion of an algorithm and convert it into a mathematically precise concept.

8.2 Turing machines

There are several mathematically precise notions of algorithm currently available. All of them represent rather complex constructions. We shall now describe one such construction, called *Turing machine* (named after a British mathematician Alan Turing). First of all, we should note that, since algorithms have to be described unambiguously, the objects on which they operate—that is, the values of the functions, and the like—must also allow unambiguous descriptions. On the other hand, all unambiguously described mathematical objects must be represented as words of an alphabet. It is, then, natural to demand that algorithm operate with words of an alphabet. That is indeed how Turing machines work.

A Turing machine is a combination of several components. For the purposes of exposition we shall not try to achieve the highest possible rigour and abstractness. The four components are as follows:

1. The *tape* which is split into a finite number of squares. The tape has direction; accordingly, it contains the leftmost and rightmost squares. The tape may change in the process, so that new squares may be attached to it from the right. Each square has squares immediately to the right and to the left of it.

2. The scanning mechanism inspecting the content of the square called the *reading head*.

3. Two finite alphabets $A = \{\#, a_0, \ldots, a_n\}$ and $Q = \{q_0, \ldots, q_t\}$, where $n, t > 0$. The elements of A are the symbols of the machine, and the elements of Q are the *states* of the machine. In each moment of time the machine is present in some state, and each square contains a certain symbol. The symbol a_0 is 'blank', q_t is the final state, and q_1 is the initial state. The symbol # prevents the scanning device from exiting the tape. For convenience we shall let $\# = a_{-1}$.

4. The machine's programme is a finite set of words, conventionally called 'commands'. They have the form:
$$q_i a_j \rightarrowtail a_k C q_l,$$
where:
   (a) $q_i \neq q_n$ and $C \in \{L, R, H\}$;
   (b) Either $a_j = a_k = \#$, or else $a_j \neq \#$ and $a_k \neq \#$.

The above command would mean that every time the machine is in the state q_i and is scanning the square with the content a_j, its next step will be as follows:

1. In the scanned square the symbol a_j is deleted and replaced by a_k;
2. The reading head moves one square to the left if $C = L$, or to the right if $C = R$, or stays put if $C = H$;
3. The machine moves into the state q_l.

The correct computation must satisfy the following conditions:

1. Only the leftmost square contains #;
2. At the beginning and at the end the reading head scans the leftmost square;
3. At the beginning the machine is present in the initial state q_1, and at the end—in the final state q_0;
4. At the end all blank squares, if there are any, are located to the right of the non-blank squares.

We can now introduce a formal notion of correct computation. Let A^* be the set of all words of the alphabet A and let $A^+ \subseteq A^*$ be the set of all non-empty words.

Definition 8.1. A word of the form xq_iy is the *situation* of the Turing machine T, where $x \in A^*$, $y \in A^+$, $q_i \in Q$, and xy is representable as $\#z$, where z is #-free.

Intuitively the situation is a snapshot of the machine at any given moment: xy is the content of the tape, q_i is the state of the machine, whereas the first letter of y reflects the square currently scanned. The situation is called *initial* if it has the form $q_1\#t$. The situation is called *final* if it has the form $q_0 u a_0^m$, where u is a_0-free. We can now refine the notion of machine command:

Definition 8.2. The situation s' is obtained from the situation s by applying the command K if one of the following conditions holds:

1. $K = q_i a_j \rightarrowtail a_k H a_l$, $s = v q_i a_j w$, $s' = v q_l a_k w$;
2. $K = q_i a_j \rightarrowtail a_k L q_l$, $s = v a_r q_i a_j w$, $s' = v q_l a_r a_k w$;
3. $K = q_i a_j \rightarrowtail a_k L q_l$, $s = v q_i a_j a_r w$, $s' = v a_k q_l a_r w$;
4. $K = q_i a_j \rightarrowtail a_k L q_l$, $s = v q_i a_j$, $s' = v a_k q_l a_0$.

(The last clause evidently refers to the case when an extension of tape is required.)

The precise definition of correct Turing computation is as follows:

Definition 8.3. A sequence $\langle s_0, s_1, \ldots, s_p \rangle$ is the *correct computation* by a Turing machine M if:

1. For every i, s_{i+1} is obtained from s_i by application of one of the commands of T;
2. s_0 is the initial state;
3. s_p is the final state.

If there is a correct computation of the machine M which starts with $q_1\#t_1a_0\cdots a_0t_n$ and finishes with the situation where $q_0\#ua_0^m$, then we say that M is processing the array $\langle t_1,\ldots,t_n\rangle$ into the word u and we shall write $u = M[t_1,\ldots,t_n]$. In this case we say that M is *applicable* to the array $[t_1,\ldots,t_n]$. In general, M is applicable to the word $W \in A_{\text{in}}^*$ if $M[W]$ stops after a finite number of steps and there is $u \in A_{\text{out}}^*$ such that $u = M[W]$.

Let $A_{\text{in}}, A_{\text{out}} \subseteq A - \{a_{-1}, a_0\}$. We can now give a definition of the Turing-computable function:

Definition 8.4. A Turing machine M *computes* the function f of the arity n if f satisfies the following conditions:

1. The domain of f is contained in A_{in}^{*n}, while its range is in A_{out}^*;
2. $\langle x_1,\ldots,x_n\rangle \in A_{\text{in}}^{*n}$ just in case if $y = M[x_1,\ldots,x_n]$, $y \in A_{\text{out}}^*$, then $y = f(x_1,\ldots,x_n)$.

A function f is *Turing-computable* if there is a Turing machine computing f.

Analogously we may define the notion of a Turing-computable predicate:

Definition 8.5. A predicate $F(x_1,\ldots,x_n)$ is *Turing-computable* if there is a Turing-computable function f such that:

$$f(x_1,\ldots,x_n) = \begin{cases} 0 & \text{if } \langle x_1,\ldots,x_n\rangle \in F \\ 1 & \text{otherwise.} \end{cases}$$

Let us now consider particular Turing machines.

Example 8.6. Let us build a machine which to each word x in the alphabet would attach the symbol a_1 from the right, so that $xa_1 = M[x]$:

$$\begin{aligned} q_1a_i &\rightarrowtail a_iRq_1 & \qquad q_1a_0 &\rightarrowtail a_1Lq_2 \\ q_2a_i &\rightarrowtail a_iLq_2 & \qquad q_2\# &\rightarrowtail \#Hq_0, \end{aligned}$$

where $i = -1, 1,\ldots,n$. We shall check the correctness of our machine by letting $x = a_1a_2a_1$. We shall then obtain the following computation:

$$\begin{aligned} &\#q_1a_2a_1a_2 \\ &\#a_2q_1a_1a_2 \\ &\#a_2a_1q_1a_2 \\ &\#a_2a_1a_2q_1a_0 \\ &\#a_2a_1q_2a_2a_1 \\ &\#a_2q_2a_1a_2a_1 \\ &\#q_2a_2a_1a_2a_1 \\ &q_2\#a_2a_1a_2a_1 \\ &q_0\#a_2a_1a_2a_1. \end{aligned}$$

Example 8.7. Let us build a machine which doubles every word $x \in A_{\text{in}}^*$, or in other words, which computes the function $f(x) = xx$. If x is empty, then we may let $f(x) = x$.

$$\begin{aligned} q_1\# &\rightarrowtail \#Rq_1 & \qquad q_1a_i &\rightarrowtail a_i'Rq_{1i} \\ q_{1i}a_j &\rightarrowtail a_j'Rq_{1i} & \qquad q_{1i}a_i'' &\rightarrowtail a_j''Rq_{1i} \\ q_{1i}a_0 &\rightarrowtail a_i''Lq_2 & \qquad q_2a_i &\rightarrowtail a_iLq_2 \\ q_2a_i'' &\rightarrowtail a_i''Lq_2 & \qquad q_2a_i' &\rightarrowtail a_i'Rq_1 \\ q_1a_i'' &\rightarrowtail a_i''Rq_3 & \qquad q_3a_i'' &\rightarrowtail a_i''Rq_3 \\ q_3a_0 &\rightarrowtail a_0Lq_4 & \qquad q_4a_i'' &\rightarrowtail a_iLq_4 \\ q_4a_i &\rightarrowtail a_iLq_4 & \qquad q_4\# &\rightarrowtail \#Hq_0 \\ q_1a_0 &\rightarrowtail a_0Lq_0, \end{aligned}$$

where $i, j = 1,\ldots,n$. Here the principle is that every symbol of the input word is copied to the right. The copied symbol is marked to avoid its repeated copying and is 'remembered' with the state. Then the reading head moves

to the right to reach the first empty square where the copy is being written. This copy is also marked (primed twice) to avoid repeated copying. Then the head moves to the left: the encounter with the primed symbol tells it to start the cycle all over again. If the cycle begins with the symbol which is marked twice, then the copying is finished. All that we have to do is to remove the marks. The final command purports to deal with an empty input word.

We should next consider the case where we need to build a machine computing the value of the functions defined on natural numbers and returning natural numbers as values. More precisely, we write ω to denote the set of natural numbers. Then we are interested in the functions $f\colon X \to \omega$, where $X \subseteq \omega^n$ for some n. Here we introduce a form of encoding natural numbers on the tape. We stipulate $A_{\text{in}} = \{|\}$ and represent the number m by a word consisting of m strokes. Such a word is sometimes called the *numeral* of the number m and is denoted by $\overline{m}$. It is then natural to define a function $\overline{f}\colon \{|\}^* \times \cdots \times \{|\}^* \to \{|\}$, such that:

$$f(x_1, \ldots, x_n) = y \iff \overline{f}(\overline{x_1}, \ldots, \overline{x_n}) = \overline{y}.$$

From the examples we have considered above it will then be clear how to build Turing machines $f(n) = n$, $f(n) = n + 1$, and $f(n) = 2n$.

8.3 Recursive functions

The already mentioned function $f\colon X \to \omega$ ($X \subseteq \omega^n$) will be called a *partial* function.

Definition 8.8. A partial function f^n is *computable* if there is an algorithm $\mathfrak{B}$ processing n-tuples $\alpha \in \omega^n$ and which does not process n-tuples $\alpha \notin X$ such that $\mathfrak{B}[\alpha] = f(\alpha)$, where $\alpha \in X$.

The following claim was put forward by Turing:

Proposition 8.9 (Turing's thesis)**.** *Every computable partial function is Turing-computable.*

Proof. None. (Discussion...) □

We shall now attempt to further refine the notion of a computable function. We shall use a popular method of calculating the values of the function f of natural numbers (*i.e.* having the same form $f\colon X \to \omega$). The main idea here is in laying down the value $f(0)$ and fixing the *recursive* formula $f(n + 1) = h(f(n))$ which allows to compute the value $f(n + 1)$ if we already know the value $f(n)$. Such a process is generally called *recursion*. The issue of computing one function is then reduced to the issue of computing another.

Example 8.10. The function $f(n) = 2^n$ can be represented as:

$$f(n) = \begin{cases} 1 & \text{if } n = 0 \\ 2^{n-1} \cdot 2 & \text{if } n > 0. \end{cases}$$

Then the function h will take the form $h(m) = m \cdot 2$.

An important fact to notice is that if a function is intuitively computable, then the function obtained from it by recursion will also be intuitively computable. In fact, we can generalise the notion of recursion a little further. The function $f(n + 1)$ may depend not only on $f(n)$, but also on n. In this case we say that $f(n + 1) = h(n, f(n))$. Thus, for $f(n) = n!$ (where $f(0) = 1$) we have $h(l, m) = m \cdot (l + 1)$. So $(n + 1)! = n! \cdot (n + 1)$. Also, we may include a rule for computing n-ary functions. If we compute a function $f(x_1, \ldots, x_k)$ and recursion is conducted on the variable x_k, then $f(x_1, \ldots, x_{k-1}, 0) = g(x_1, \ldots, x_{k-1})$, while $f(x_1, \ldots, x_k + 1) = h(x_1, \ldots, x_{k-1}, x_k, f(x_1, \ldots, x_{k-1}, x_k))$, where g^{k-1} and h^{k+1} are computable functions. (Intuitively, g expresses the initial condition, and h is the recursive step.)

Recursion is not unique in having a computability-preserving property. Another way is *superposition*, whereby our ability to compute $f(x)$ and $g(x)$ entails the ability to compute $g(f(x))$. Now, if we have a stock of simple functions which could be computed in a trivial way, then the functions obtained from them by recursion and superposition would also be computable. The *basic functions* include the following:

1. The zero function $Z(x) = 0$ for every x;
2. The successor function $S(x) = x + 1$ for every x;
3. The projection function $I_i^n(x_1, \ldots, x_n) = x_i$ for every $x_1, \ldots, x_n$, where $1 \leq i \leq n$.

Definition 8.11. A function f is *primitive recursive* if either f is one of the basic functions, or else it is obtained from them by recursion and superposition.

Some instances of primitive recursive functions include:

1. $Z^n(x_1, \ldots, x_n) = Z(I_i^n(x_1, \ldots, x_n))$.
2. Constant functions of the form $k(x)$: $1(x) = S(Z(x))$, $2(x) = S(1(x))$, and so forth.
3. The function $f(x_1, x_2) = x_1 + x_2$ is obtained by recursion from $I_1^1(x_1)$ and $S(I_3^3(x_1, x_2, x_3)) = x_3 + 1$, so that we have:
$$\begin{cases} x_1 + 0 = I_1^1(x_1) \\ x_1 + (x_2 + 1) = (x_1 + x_2) + 1. \end{cases}$$
4. The function of limited subtraction:
$$x \dot{-} y = \begin{cases} x - y & \text{if } x \geq y \\ 0 & \text{otherwise} \end{cases}$$
is obtained as follows:
$$\begin{cases} x \dot{-} 0 = x \\ x \dot{-} (y + 1) = (x \dot{-} y) \dot{-} 1. \end{cases}$$

Predicates are treated analogously. Let $F(x_1, \ldots, x_n)$ be an n-ary predicate. We define the function χ_F^n as follows:
$$\chi_F(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } \langle x_1, \ldots, x_n \rangle \in F \\ 0 & \text{if } \langle x_1, \ldots, x_n \rangle \notin F. \end{cases}$$

The function χ_F is then called the *characteristic function* of the predicate F.

Definition 8.12. The predicate $F(x_1, \ldots, x_n)$ defined on ω is *primitive recursive* if its characteristic function χ_F is primitive recursive.

[Men64, BJ89]

Chapter 9

Incompleteness results for arithmetic

9.1 Set-theoretic and semantic paradoxes

...

9.2 A Gödelian puzzle

We consider a machine M (perhaps a Turing machine) which prints out various expressions. Let the alphabet $A_{\text{out}} = \{\neg, P, N, (,)\}$. An expression $X \in A^*_{\text{out}}$ is *printable* if M prints it. The expression $X(X)$ is called *the norm* of the expression X. Clearly $X(X) \in A^*_{\text{out}}$. A *sentence* will be an expression of one of the four forms:

1. $P(X)$
2. $PN(X)$
3. $\neg P(X)$
4. $\neg PN(X)$.

We then let $V(P(X)) = \mathbf{1}$ iff X is printable, and $V(PN(X)) = \mathbf{1}$ iff the norm of X is printable. Similarly for the rest.

We now obtain a case of self-reference, since our machine M can print various sentences that say in effect what the machine can or cannot print. On the other hand, if M ever prints $P(X)$, then X is really printable. And if $PN(X)$ is printable, then $X(X)$ is also printable.

Suppose X is printable. Does it follow that $P(X)$ is printable? No. If X is printable, then $P(X)$ is true. But we have not shown that *all* true sentences must be printed by M. All we know is that M does not print any false sentences. Let us, therefore, ask whether M could possibly print all true sentences. The answer is again negative. There is at least one true sentence not printable by M. Consider the sentence $\neg PN(\neg PN)$. It is true just in case the norm of the expression $\neg PN$ is not printable. But the norm of $\neg PN$ is exactly $\neg PN(\neg PN)$. Hence $\neg PN(\neg PN)$ is true iff it is not printable. Since M prints no false sentences, the remaining option is that the sentence is true, but is not printable.

9.3 The abstract forms of Gödel's and Tarski's theorems

Let us consider a broad notion of mathematical theory to which Gödel's argument is applicable. To this end let us fix the components of the signature Σ of such a theory.

1. A countably infinite set $\mathscr{E}$ of expressions.
2. A set $\mathscr{S} \subseteq \mathscr{E}$ of sentences.
3. A set $\mathscr{P} \subseteq \mathscr{S}$ of provable sentences.
4. A set $\mathscr{R} \subseteq \mathscr{S}$ of refutable sentences.
5. A set $\mathscr{T} \subseteq \mathscr{S}$ of true sentences.

6. A set $\mathscr{H}$ of predicates. (Generally, predicates are expressions. Informally, they may be thought as sets of natural numbers.)

7. A function ϕ assigning to each expression E and every natural number n an expression $E(n)$ such that, for $H \in \mathscr{H}$, $H(n)$ is a sentence. (Informally, $H(n)$ says that the number n belongs to the set H.)

We say that a predicate H is true for $n \in \omega$ (or is *satisfied* by n) if $H(n) \in \mathscr{T}$. The set *expressed* by H is the set of all $n \in \omega$ satisfied by H.

Definition 9.1 (Expressibility)**.** Let X be a set of natural numbers. H *expresses* X iff for every number n, $H(n) \in \mathscr{T} \leftrightarrow n \in X$. The set X is called *expressible* in Σ if there is a predicate of Σ which expresses X.

Since the set $\mathscr{E}$ is countably infinite, there are at most denumerably many predicates of Σ. A question arises whether all sets of natural numbers would be expressible in Σ. That would be the case if there were denumerably many sets of natural numbers. That this is not so is a consequence of the following celebrated theorem. (Here the reader is encouraged to consult §2.2.2 for definitions of key notions.)

Proposition 9.2 (Cantor's theorem)**.** *The set $\wp(X)$ of all subsets of X has the cardinality strictly greater than the cardinality of X.*

Proof. First we show that $|X| \leq |\wp(X)|$. This is done by associating with each $x \in X$ a set $\{x\} \in \wp(X)$. This defines a one-to-one mapping of X into $\wp(X)$.

Suppose, for *reductio*, that $|X| = |\wp(X)|$ and that, therefore, there exists a one-to-one mapping f of X *onto* $\wp(X)$. Consider the set $M = \{x \in X \mid x \notin f(x)\}$. Clearly $M \subseteq X$. Then $M \in \wp(X)$. And then, by assumption, there must be $m \in X$ such that $f(m) = M$. Hence, on the one hand, if $m \in M$, then $m \notin f(m) = M$, and, on the other hand, if $m \notin M$, then $m \in f(m) = M$. We have obtained a contradiction. □

Now, let ω be the set of natural numbers. Then the set $\wp(\omega)$ of its subsets (*i.e.* the set of the sets of natural numbers) will have a cardinality greater than $|\omega|$. Therefore, there are more than denumerably many sets of natural numbers, and therefore, not every set of numbers will be expressible in Σ.

Consider now a theory T of the signature Σ.

Definition 9.3. A theory T is *correct* if its every sentence $A \in \Sigma$ provable in T is true, and every sentence $A \in \Sigma$ refutable in T is false.

In a correct theory, in other words, $\mathscr{P} \subseteq \mathscr{T}$, while $\mathscr{R} \cap \mathscr{T} = \varnothing$. We are now going to show that a correct theory T must contain a true sentence not provable in T.

9.3.1 The diagonal method

Let g be a one-to-one injection assigning to each expression E a number. That number $g(E)$ is called the *Gödel number* of E. We also assume that for every number there is an expression which has that number as its Gödel number. We further let E_n be an expression such that $g(E_n) = n$.

The *diagonalisation* of E_n is the expression $E_n(n)$. Given that E_n is a predicate, the sentence $E_n(n)$ is true iff the predicate E_n is satisfied by its own Gödel number n. Further, let $d(n)$ be the Gödel number of $E_n(n)$.

Definition 9.4 (Diagonal sets)**.** Let X be a set of numbers. The *diagonal set* X_d will be the set of all n such that $d(n) \in X$. In other words:

$$n \in X_d \leftrightarrow d(n) \in X.$$

9.3.2 Gödel's theorem: an abstract form

Definition 9.5. The set P is the set of Gödel numbers of all provable sentence:

$$x \in P \leftrightarrow x = g(A) \wedge A \in \mathscr{P}.$$

Definition 9.6. The *complement* of a set $X \subseteq \omega$ is the set $\widetilde{X}$ containing all natural numbers not in X:

$$\widetilde{X} = \{x \in \omega \mid x \notin X\}.$$

(Notice the deviation from our notation in Chapter 2.)

We are now ready to state Gödel's theorem in a fairly general form (*i.e.* without imposing specific restrictions on T).

Proposition 9.7 (After Gödel)**.** *If the set $\widetilde{P}_d$ is expressible in T and T is correct, then there is a sentence $A \in \Sigma$ such that $A \in \mathscr{T}$ and $A \notin \mathscr{P}$.*

Proof.

T is correct.	Ass.	(1)
$\widetilde{P}_d$ is expressible in T.	Ass.	(2)
$\forall n(H(n) \in \mathscr{T} \leftrightarrow n \in \widetilde{P}_d)$	Def. 9.1, (2)	(3)
$h = g(H)$	Ass.	(4)
$H(h) \in \mathscr{T} \leftrightarrow h \in \widetilde{P}_d$	UI, (3)	(5)
$h \in \widetilde{P}_d \leftrightarrow d(h) \in \widetilde{P} \leftrightarrow d(h) \notin P$	Def. 9.4, Def. 9.6	(6)
$d(h) = g(H(h))$	(4)	(7)
$d(h) \in P \leftrightarrow H(h) \in \mathscr{P}$	Def. 9.5, (7)	(8)
$d(h) \notin P \leftrightarrow H(h) \notin \mathscr{P}$	(8), PL	(9)
$H(h) \in \mathscr{T} \leftrightarrow H(h) \notin \mathscr{P}$	(5), (6), (9)	(10)
$(H(h) \in \mathscr{T} \wedge H(h) \notin \mathscr{P}) \vee (H(h) \notin \mathscr{T} \wedge H(h) \in \mathscr{P})$	(10), PL	(11)
$(H(h) \notin \mathscr{T} \wedge H(h) \in \mathscr{P}) \supset (\mathsf{T}$ is not correct$)$	Def. 9.3	(12)
$\neg(H(h) \notin \mathscr{T} \wedge H(h) \in \mathscr{P})$	(12), (1), PL	(13)
$H(h) \in \mathscr{T} \wedge H(h) \notin \mathscr{P}$	(11), (13), PL	(14)

Therefore, $H(h)$ is the desired sentence. □

The following important notion is implicitly used in the above proof.

Definition 9.8 (Gödel sentences)**.** Let $X \subseteq \omega$. The *Gödel sentence* E_n for X is a sentence with the following property:

$$E_n \in \mathscr{T} \leftrightarrow n \in X.$$

(Informally, a Gödel sentence asserts that its own Gödel number lies in X.)

When considering a theory T (of the signature Σ) with specific properties it will be convenient to verify the claim that $\widetilde{P}_d$ is expressible in T by verifying three conditions:

G_1**:** For any set X expressible in T, the set X_d is expressible in T.

G_2**:** For any set X expressible in T, the set $\widetilde{X}$ is expressible in T.

G_3**:** The set P is expressible in T.

The three conditions jointly imply that $\widetilde{P}_d$ is expressible in T. It is in general the third condition which demands greater effort.

9.3.3 Tarski's theorem: an abstract form

A particularly simple route to Gödel's incompleteness theorems is offered by Tarski's theorem on the indefinability of truth. We shall now state it, again, without making specific assumptions about T. We first prove the following lemma.

Proposition 9.9 (Diagonal lemma)**.** *For any set X, if X_d is expressible in T, then there is a Gödel number for X.*

Proof.

H expresses X_d in T.	Ass.	(15)
$\forall n(H(n) \in \mathscr{T} \leftrightarrow n \in X_d)$	Def. 9.1, (15)	(16)
$h = g(H)$	Ass.	(17)
$d(h) = g(H(h))$	(17)	(18)
$H(h) \in \mathscr{T} \leftrightarrow h \in X_d$	UI, (16)	(19)
$h \in X_d \leftrightarrow d(h) \in X$	Def. 9.4, Def. 9.6	(20)
$H(h) \in \mathscr{T} \leftrightarrow d(h) \in X$	(19), (20)	(21)
$H(h)$ is a Gödel sentence for X.	(18), (21), Def. 9.8	□

Proposition 9.10. *If* T *satisfies the condition* G_1*, then for any* X *expressible in* T*, there is a Gödel number for* X*.*

Proof. If X is expressible in T, then, by G_1, X_d is expressible in T. Then, by the Diagonal lemma, there is a Gödel number for X. □

The abstract form of Gödel's theorem above is now allowed an even shorter proof.

Proof of Gödel's theorem based on the Diagonal lemma.

T is correct.	Ass.	(22)
$\widetilde{P}_d$ is expressible in T.	Ass.	(23)
There is a Gödel sentence G for $\widetilde{P}$.	(23), Diagonal lemma	(24)
$G \in \mathscr{T} \leftrightarrow G \notin \mathscr{P}$	(24), Def. 9.5	(25)
$G \in \mathscr{T} \wedge G \notin \mathscr{P}$	(22), (25), PL	□

The Diagonal lemma also yields Tarski's theorem.

Proposition 9.11 (After Tarski)**.** *Let* T *be the set of Gödel numbers of the true sentences of* T*. Then the following claims hold:*

1. *The set* $\widetilde{T}_d$ *is not expressible in* T*.*
2. *If* G_1 *holds for* T*, then* $\widetilde{T}$ *is not expressible in* T*.*
3. *If* G_1 *and* G_2 *hold for* T*, then* T *is not expressible in* T*.*

Proof. We shall prove each of the claims in its own turn.

1. If $\widetilde{T}_d$ were expressible in T, then, by the Diagonal lemma, there must be a Gödel sentence for $\widetilde{T}$. But such a sentence would have been true iff its Gödel number was not the Gödel number of a true sentence. This is impossible; therefore, there is no Gödel sentence for $\widetilde{T}$. Hence, $\widetilde{T}_d$ is not expressible in T.
2. Suppose G_1 holds. Then, if $\widetilde{T}$ were expressible in T, $\widetilde{T}_d$ must be expressible in T. But this contradicts the just proven claim 1.
3. Suppose G_1 and G_2 hold. Then, if T were expressible in T, $\widetilde{T}$ must be expressible in T. But this contradicts the just proven claim 2. □

9.4 Tarski's theorem for arithmetic

There are several ways of proving incompleteness of Peano arithmetic—a standard axiomatisation of arithmetic. Closely following [Smu94], we are now interested in an especially simple proof utilising Tarski's theorem.

9.4.1 Syntax

We now move on to investigate a first-order arithmetical theory involving addition, multiplication, and exponentiation. The alphabet that we use includes the following thirteen symbols:

$$0 \quad ' \quad (\quad) \quad f \quad \prime \quad v \quad \neg \quad \supset \quad \forall \quad = \quad \leq \quad \sharp$$

The expressions 0, $0'$, $0''$, and so forth, are called *numerals* and will be used as names of the numbers 0, 1, 2, and so forth. It is then clear that $'$ serves as a name of the successor function. The symbols $f\prime$, $f\prime\prime$, and $f\prime\prime\prime$ are the names of the operations of addition, multiplication, and exponentiation. They are abbreviated as $+$, $\cdot$, and $\mathbf{E}$ respectively. The usual logical constants of the first-order calculus retain their meaning. But we also need a countably infinite list of variables $v_1, v_2, \ldots, v_n, \ldots$; these we put in our 13-symbol alphabet by abbreviating $v_1, v_2, v_3, \ldots$ as $v\prime, v\prime\prime, v\prime\prime\prime, \ldots$.

Terms are formed according to the following rules:

1. Every variable and numeral is a term.
2. If t_1 and t_2 are terms, then $(t_1 + t_2)$, $(t_1 \cdot t_2)$, and $(t_1 \mathbf{E} t_2)$ are also terms.

A *constant* term contains no free variables.

An *atomic formula* is expression of the form $t_1 = t_2$ and $t_1 \leq t_2$. The set of formulae is formed according to the usual rules of first-order calculus.

The notions of *free* and *bound* variables and of *sentences* carry over from the first-order calculus. A special comment is needed for *substitution* of numerals for variables. For any number n, by $\underline{n}$ we mean the numeral designating n (this accords with our earlier notation for designating elements in a model). Thus, $\underline{5}$ abbreviates the expression $0''''''$. Further, deviating somewhat from our previous notation, we write $A(v_i)$ to indicate a formula having v_i as its only free variable. Similarly, for any numbers $k_1, \ldots, k_n$, we write $A(\underline{k_1}, \ldots, \underline{k_n})$ to mean a formula obtained by substituting the numerals $\underline{k_1}, \ldots, \underline{k_n}$ for the free occurrence of the variables $v_{i_1}, \ldots, v_{i_n}$. A formula is said to be *regular* if $i_1 = 1, \ldots, i_n = n$.

The *complexity* of formulae is interpreted analogously to the case of first-order calculus.

The following abbreviations will be used:

$$\begin{aligned}
(A \vee B) &\Longleftrightarrow (\neg A \supset B) \\
(A \wedge B) &\Longleftrightarrow \neg(A \supset \neg B) \\
A \leftrightarrow B &\Longleftrightarrow ((A \supset B) \wedge (B \supset A)) \\
\exists v_i A &\Longleftrightarrow \neg \forall v_i \neg A \\
t_1 \neq t_2 &\Longleftrightarrow \neg t_1 = t_2 \\
t_1 < t_2 &\Longleftrightarrow ((t_1 \leq t_2) \wedge (t_1 \neq t_2) \\
t_1^{t_2} &\Longleftrightarrow t_1 \mathbf{E} t_2 \\
(\forall v_i \leq t) A &\Longleftrightarrow \forall v_i (v_i \leq t \supset A) \\
(\exists v_i \leq t) A &\Longleftrightarrow \neg(\forall v_i \leq t) \neg A.
\end{aligned}$$

We shall also omit outward parentheses, in accordance with our earlier usage. Another auxiliary notion is that of *designation*. It is defined according to the following rules:

1. A numeral $\underline{n}$ designates the number n.
2. If the constant terms c_1 and c_2 designate the numbers n_1 and n_2, then $(c_1 + c_2)$ designates the sum $n_1 \dot{+} n_2$, $(c_1 \cdot c_2)$ designates the product $n_1 \dot{\cdot} n_2$, $(c_1 \mathbf{E} c_2)$ designates the number $n_1^{n_2}$, and c_1' designates $n_1 + 1$.

9.4.2 The notion of truth

The definition of truth is carried out by induction on the complexity of formulae.

T_0**:** An atomic sentence $c_1 = c_2$ is true iff c_1 and c_2 designate the same natural numbers. An atomic sentence $c_1 \leq c_2$ is true iff c_1 designates the number less or equal than the number designated by c_2.

T_1**:** A sentence $\neg A$ is true iff A is not true.

T_2: A sentence $A \supset B$ is true iff either A is not true, or B is true.

T_3: A sentence $\forall v_i F$ is true iff for every $n \in \omega$, the sentence $F(\underline{n})$ is true.

An open formula $F(v_{i_1}, \ldots, v_{i_k})$ is *correct* if for all numbers $n_1, \ldots, n_k$, the sentence $F(\underline{n_1}, \ldots, \underline{n_k})$ is true.

Substitution and equivalence are defined in the usual way.

9.4.3 The notion of Arithmetic and arithmetic sets and relations

We can generalise our earlier notion of expressibility. We say that $F(v_1, \ldots, v_n)$ expresses the set of n-tuples $\langle k_1, \ldots, k_n \rangle$ (where $k_i \in \omega$). That is, $F(v_1, \ldots, v_n)$ *expresses* the relation $R(x_1, \ldots, x_n)$ just in case:

$$F(\underline{k_1}, \ldots, \underline{k_n}) \text{ is true } \leftrightarrow R(k_1, \ldots, k_n).$$

Example 9.12. The formula $\exists v_2(v_1 = 0'' \cdot v_2)$ expresses the set of even numbers.

A set or relation is called *Arithmetic* if it is expressed by a formula of $\mathscr{L}_E$. A set or relation is called *arithmetic* if it is expressed by a formula of $\mathscr{L}_E$ in which the exponential symbol does not occur. Analogously, a function $f(x_1, \ldots, x_n)$ is Arithmetic iff there is a formula $F(v_1, \ldots, v_n, v_{n+1})$ such that for all $x_1, \ldots, x_n, y \in \omega$, the sentence $F(\underline{x_1}, \ldots, \underline{x_n}, \underline{y})$ is true iff $f(x_1, \ldots, x_n) = y$.

9.4.4 Concatenation

We are looking to define the function $x *_b y$ for any $b > 1$.

Example 9.13. $42 *_{10} 8768 = 428768$.

We notice that $m *_{10} n = m \cdot 10^{\ell(n)} + n$. Hence, generally, $m *_b n = m \cdot b^{\ell(n)} + n$.

Proposition 9.14. *For every $b > 1$, the relation $x *_b y = z$ is Arithmetic.*

Proof. Omitted. □

Proposition 9.15. *For every $n, b > 1$, the relation $x_1 *_b \ldots *_b x_n = y$ is Arithmetic.*

Proof. By induction on n. □

9.4.5 Gödel numbering

We assign Gödel numbers to expressions in order to be able to talk about expressions 'indirectly' by talking about their Gödel numbers.

The idea of [Qui51] was to use the base 10 notation so that the expression $S_5 S_3 S_4$ were assigned the Gödel number 534. We use the base 13 notation and some modifications will be required. We use η, ϵ, and δ as digits for 10, 11, and 12 respectively. Thus we assign Gödel numbers to our thirteen symbols as follows:

0	$'$	$($	$)$	f	$\prime$	v	$\neg$	$\supset$	$\forall$	$=$	$\leq$	$\sharp$
1	0	2	3	4	5	6	7	8	9	η	ϵ	δ

Example 9.16. The Gödel number of the expression $0'' \leq 0'''$ is the number $100\epsilon1000_{13}$, that is, the number $0 + 0 \cdot 13 + 0 \cdot 13^2 + 1 \cdot 13^3 + 11 \cdot 13^4 + 0 \cdot 13^5 + 0 \cdot 13^6 + 1 \cdot 13^7$.

For any two expressions E_x and E_y the Gödel number of the expression $E_x E_y$ is $x *_{13} y$.

Notice also that the numeral $\underline{n}$ consists of 0 followed by n primes, so that its Gödel number consists of 1 followed by n occurrences of n. Hence it is 13^n.

9.4.6 Tarski's theorem

The notions of Gödel sentence carries over from the earlier discussion.

Given any formula $F(v_1)$, the sentence $F(\underline{n})$ is equivalent to the sentence $\forall v_1(v_1 = \underline{n} \supset F(v_1))$. We shall designate $\forall v_1(v_1 = \underline{n} \supset F(v_1))$ as $F[\underline{n}]$. (The same abbreviation holds for any expression E which is not necessarily a formula.)

Let $e, n \in \omega$. Let the Gödel number of E be e. Then by $r(e, n)$ we mean the Gödel number of the expression $E[\underline{n}]$. Our goal is to show that $r(x, y)$ is Arithmetic. (The function $r(x, y)$ is also called the *representation function* of $\mathscr{L}_E$.)

Let $E_x[\underline{y}]$ be the expression $\forall v_1(v_1 = \underline{y} \supset E_x)$. Suppose the Gödel number of $\forall v_1(v_1 = \text{is } k$. Then the Gödel number of $E_x[\underline{y}]$ is $k * 13^y * 8 * x * 3$. So we can see that $r(x, y)$ can be written as $\exists w(w = 13^y \wedge z = k * w * 8 * x * 3$. Therefore:

Proposition 9.17. *The function $r(x, y)$ is Arithmetic.*

9.4.7 Diagonalisation and Tarski's theorem

Let the function $d(x) = r(x, x)$. The function d is called the *diagonal function*. Clearly it is Arithmetic, since r is Arithmetic. For any $n \in \omega$, $d(n)$ will be the Gödel number of $E_n[\underline{n}]$. We can now introduce the already familiar notion of the *diagonal set*. For any $X \subseteq \omega$, X_d is the set of n such that $d(n) \in X$.

Proposition 9.18. *If X is Arithmetic, then so is X_d.*

Proof. The set X_d consists of x such that $\exists y(d(x) = y \wedge y \in X)$. Since $d(x)$ is Arithmetic, there is $D(v_1, v_2)$ expressing the relation $d(x) = y$. But suppose $F(v_1)$ expresses X. Then X_d is expressed by $\exists v_2(D(v_1, v_2) \wedge F(v_2))$. □

We can now prove the following claim to be used (later ...) in constructing a true unprovable sentence:

Proposition 9.19. *If X is Arithmetic, then there is a Gödel sentence for X.*

Proof. Suppose X is Arithmetic. Then X_d is Arithmetic, too (by Proposition 9.18). Let $H(v_1) \in \mathscr{L}_E$ be a formula expressing X_d, and let h be its Gödel number. Then:

$$H(v_1) \text{ is true } \leftrightarrow h \in X_d \leftrightarrow d(h) \in X.$$

But $d(h)$ is the Gödel number of $H[\underline{h}]$. Therefore, $H[\underline{h}]$ is a Gödel sentence for X. □

We can now prove Tarski's theorem.

Proposition 9.20 (Tarski)**.** *The set T of Gödel numbers of the true Arithmetic sentences is not Arithmetic.*

Proof. We first show that $\widetilde{T}$ is not Arithmetic. Indeed, if $\widetilde{T}$ were Arithmetic, then, by Proposition 9.19, there would be a Gödel sentence for $\widetilde{T}$. But there is no Gödel sentence for $\widetilde{T}$, since such a sentence would be true if and only if it were not true. So $\widetilde{T}$ is not Arithmetic.

On the other hand, if $F(v_1)$ expresses X, then $\neg F(v_1)$ expresses $\widetilde{X}$. Hence, for every X, if X is Arithmetic, then $\widetilde{X}$ is Arithmetic, too. And since $\widetilde{T}$ is not Arithmetic, it now follows that T is not Arithmetic. □

Armed with Tarski's theorem, we will be able to argue that, since the set of Gödel numbers of all provable sentences is Arithmetic, truth and provability do not coincide.

9.5 Peano arithmetic

9.6 Gödel's theorems

9.7 Further notes on undecidability and incompleteness

[Smu94]

Chapter 10

Beginning model theory

10.1 Further properties of algebraic systems

10.2 Compactness, categoricity

10.3 The Skolem-Löwenheim theorem

10.4 Craig's interpolation theorem and its applications

[Smu68, Men64]

Chapter 11

Exercises

11.1 Revision: paraphrase etc.

Your answers must be as detailed as possible. Simple yes/no answers do not count.

11.1.1. Give the best paraphrases of the following sentences into the language of propositional logic:

1. Obama and Palin will not both win the election.
2. Either Obama will win the election and the world be saved or else he will not and Palin will triumph.
3. Nadal will win the next Grand Slam unless Federer improves.
4. If the government either prints money and creates inflation or improves the business environment, unemployment will begin to drop.
5. Barack will marry Sarah in January if he gets the divorce from Michelle in December, provided they don't kill each other before and the Democrats lose the election.
6. Barack will never marry Sarah, even if he falls in love with her.

Solution.

1. $\neg(P \wedge Q)$
2. $[(P \wedge Q) \wedge \neg R] \vee (\neg P \wedge R)$
3. $P \leftrightarrow \neg Q$
4. $[(P \wedge Q) \vee R] \supset S$
5. $(P \wedge Q) \supset (R \supset S)$
6. P

11.1.2. Could there be valid arguments under the following conditions:

1. Premisses all true, conclusion false.
2. Premisses all untrue, conclusion untrue.
3. Some premisses true, some untrue, conclusion untrue.
4. Premisses all true, negation of conclusion true.

Solution.

1. No.
2. Yes.
3. Yes.

4. No.

11.1.3. Establish whether the following arguments are valid:

1. Premisses: 'Federer is tall', 'If Federer is tall, he is handsome', 'Federer is short'. Conclusion: 'Federer is handsome.'
2. Premisses: 'Federer is tall', 'If Federer is tall, he is handsome'. Conclusion: '$2 + 2 = 4$.'
3. Premisses: 'Federer is tall', 'If Federer is tall, he is handsome'. Conclusion: 'Federer is tall.'
4. Premisses: 'Federer is tall', 'If Federer is tall, he is handsome'. Conclusion: 'Federer is short.'

Solution.

1. Valid.
2. Valid.
3. Valid.
4. Invalid.

11.2 Paraphrase/interpretation

11.2.1. Paraphrase the following sentences into a predicate language. Explain the difficulties. Indicate the interpretation of predicates and individual parameters and specify the domain for successful paraphrases.

1. Only you can help me.
2. If Harry Potter gets angry at you, you can only pray to God.
3. Boys love girls, but girls love their mothers.
4. Bill Gates is the richest man in the world.
5. Two soldiers died in that battle.
6. There are at most two students in this class.
7. There are exactly two students in this class.

Solution.

1. $\forall x(Hxa \supset x = b)$, domain: all people.
2. $Aha \supset \forall x(Dax \supset x = p)$, domain: everything.
3. $\forall x[Bx \supset \exists y(Gy \wedge Lxy)] \wedge \forall x[Gx \supset \exists y(Myx \wedge Lxy)]$, domain: all people.
4. $\neg\exists x(Mx \wedge Rxg)$, domain: people in the world (on Earth?).
5. $\exists x\exists y(Sx \wedge Sy \wedge Dxb \wedge Dyb \wedge x \neq y)$, domain: people and battles.
6. We deny that there are (at least) three students:
   $\neg\exists x\exists y\exists z(Sxc \wedge Syc \wedge Szc \wedge x \neq y \wedge x \neq z \wedge y \neq z)$, domain: people and classes.
7. $\exists x\exists y(Sxc \wedge Syc \wedge x \neq y \wedge \forall z(Szc \supset (x = z \vee y = z))$, domain: people and classes.

11.2.2. By finding relevant interpretations, establish whether the following formulae are valid/invalid and satisfiable/unsatisfiable:

1. $\forall x(Px \supset Qx) \vee \neg(\forall xPx \supset \forall xQx)$
2. $\forall y\exists x(Pxy \vee Pyx) \supset (\forall x\exists yPxy \vee \forall x\exists yPyx)$
3. $\forall y(\exists zPyz \supset \exists zPzy) \supset \forall y(\forall zPyz \supset \forall zPzy)$

Solution.

1. Invalid and satisfiable. The formula is false on the left-hand interpretation and true on the right-hand one:

$M : \{1, 2, 4, 6 \ldots\}$	$M : \{0, 2, 4, \ldots\}$
$P :$ ① is even	$P :$ ① is odd
$Q :$ ① is odd	$Q :$ ① is even.

2. Invalid and satisfiable. The formula is false on the left-hand interpretation and true on the right-hand one:

$M : \{0, 1, 2\}$	$M : \{0, 1, 2\}$
$P :$ ① $>$ ②	$P :$ ① $=$ ②.

3. Invalid and satisfiable. Note that $\neg\forall y(\forall zPyz \supset \forall zPzy) \simeq \exists y(\forall zPyz \wedge \exists w\neg Pwy)$. The formula is false on the left-hand interpretation and true on the right-hand one:

$M : \{0, 1, 2, \ldots\}$	$M : \{0, 1, 2, \ldots\}$
$P :$ ① $\leq$ ②	$P :$ ① $<$ ②.

11.3 General concepts

11.3.1. Accept or reject with a short explanation the following claims:

1. $X \subseteq X$
2. $X \subset X$
3. $\{a, b\} \in \{a, b, c\}$

Solution. Easy.

11.3.2. Examine whether $|X| = |Y|$:

1. $X = \{0, 1, 2, 3\}$, $Y = \{5, 1, 7, 8, 5\}$
2. $X = \{0, 1, 2, \ldots\}$, $Y = \{-1, 0, 1, 2 \ldots\}$
3. $X = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$, $Y = \{\ldots, -5, -3, -1, 1, 3, 5 \ldots\}$

Solution.

1. Easy.
2. Easy.
3. The bijective mapping is $f(n) = 2n - 1$.

11.3.3. Insert missing quotation marks in the following passage:

> 'You are sad,' the Knight said in an anxious tone: 'let me sing you a song to comfort you.'
>
> 'Is it very long?' Alice asked, for she had heard a good deal of poetry that day.
>
> 'It's long,' said the Knight, 'but very, very beautiful. Everybody that hears me sing it – either it brings the tears into their eyes, or else – '
>
> 'Or else what?' said Alice, for the Knight had made a sudden pause.
>
> 'Or else it doesn't, you know. The name of the song is called Haddocks' Eyes.'
>
> 'Oh, that's the name of the song, is it?' Alice said, trying to feel interested.
>
> 'No, you don't understand,' the Knight said, looking a little vexed. 'That's what the name is called. The name really is The Aged Aged Man.'
>
> 'Then I ought to have said That's what the song is called?' Alice corrected herself.
>
> 'No, you oughtn't: that's quite another thing! The song is called Ways and Means: but that's only what it's called, you know!'
>
> 'Well, what is the song, then?' said Alice, who was by this time completely bewildered.
>
> 'I was coming to that,' the Knight said. 'The song really is A-sitting On A Gate: and the tune's my own invention.'

What is the song called? (Give a full answer.)
Solution. 'The Aged Aged Man' (the name), 'Ways and Means' (the nickname), and possibly also 'A-sitting On A Gate'.

11.3.4. Is $(P_1 \supset P_2)$ a formula of $\mathsf{H_s}$, or is '$(P_1 \supset P_2)$' a formula of $\mathsf{H_s}$?
Solution. $(P_1 \supset P_2)$ is a formula of $\mathsf{H_s}$, but is '$(P_1 \supset P_2)$' is a name of the formula of $\mathsf{H_s}$.

11.3.5. Determine whether the formula $\neg A \supset (A\neg \supset B)$ is a formula of $\mathsf{H_s}$.
Solution. Easy.

11.3.6. Four students, Abe, Ben, Chris, and Dave, competed in a table-tennis tournament. When a journalist asked them which places they got, he received three different answers:

1. Chris was first, Ben second.
2. Chris was second, Dave third.

3. Abe was second, Dave fourth.

In each of the answers at least one part is correct. None of the students got the same place. Determine who got which place.

Solution. We need to know in what situation the sentence $(c_1 \vee b_2) \wedge (c_2 \vee d_3) \wedge (a_2 \vee d_4)$ is true, given the assumptions (e.g. that there can't be two people getting to the same place). Examining the truth-table of our sentence, we see that it is true when c_1, a_2, d_3 are true. Consequently, Chris was first, Abe second, Dave third, and Ben fourth.

11.4 Sentence calculus

11.4.1. Formulate the rules for building syntactic formation trees for the formulae of $\mathsf{H_s}$ (of the kind we discussed at the lecture).
Solution. Let $\mathscr{T}$ be a formation tree for A. Then $\mathscr{T}$ is an ordered dyadic tree whose points are formulae and whose origin is A. The following conditions are imposed on $\mathscr{T}$:

1. Each end point is a sentence parameter.
2. Each simple point has the form $\neg B$, and its only successor is B.
3. Each junction point has the form $B \triangle C$, whose successors are B and C (where $\triangle$ is one of the sentential connectives excluding negation).

11.4.2. Show that:

1. If $\Gamma \cup \{\neg A\} \vdash \neg B$, then $\Gamma \cup \{B\} \vdash A$.
2. If $\Gamma \vdash A$, then $\Gamma \cup \Delta \vdash A$.
3. If $\Gamma \vdash A$ and $\Delta \cup \{A\} \vdash B$, then $\Gamma \cup \Delta \vdash B$.

Solution.

1. Using the Deduction theorem:
$$\begin{aligned}&\Gamma \cup \{\neg A\} \vdash \neg B\\ &\Gamma \vdash \neg A \supset \neg B\\ &\vdash \neg A \supset \neg B \supset (B \supset A)\\ &\Gamma \vdash B \supset A\\ &\Gamma \cup \{B\} \vdash A\end{aligned}$$
2. By the definition of deduction.
3. Using the definition of deduction and Ex. 2.

11.4.3. Are the following sets of sentences simultaneously satisfiable:

1. Mary is a woman. Mary has forty biological children.
2. Mary is a woman. Mary has three billion biological children.
3. If Padua is where I think it is, Milan is west of it. Milan is not where I think it is, but Padua is. Milan is north west of Padua.
4. Padua is in Italy or Greece. If Padua is in Greece, Italy is not in Europe. Italy is in Europe. Padua is not in Europe.
5. If Padua is in Europe, it is in Italy. Padua is in Italy, but not in Europe.

Solution.

1. Simultaneously satisfiable.
2. Controversial.
3. $X = \{P \supset Q, P \wedge \neg R, \neg Q\}$: not simultaneously satisfiable.
4. $X = \{P \vee Q, Q \supset \neg R, R, \neg S\}$. But one might also assume that $P \supset S$ (why?). Then we would have $X' = \{P \vee Q, Q \supset \neg R, R, \neg S, P \supset S\}$. X' is not simultaneously satisfiable, but X is.
5. $X = \{P \supset Q, Q \wedge \neg P\}$: simultaneously satisfiable.

11.4.4. Find a disjunctive normal form for the formula $(Q \wedge R) \supset (P \leftrightarrow (\neg Q \vee R))$.
Solution. $P \vee \neg Q \vee \neg R$.

References

[BJ89] G. Boolos and R. Jeffrey. *Computability and Logic*. Routledge & Kegan Paul, 1989.

[Bos97] D. Bostock. *Intermediate Logic*. Oxford University Press, 1997.

[Men64] E. Mendelson. *Introduction to Mathematical Logic*. Van Nostrand Company, 1964.

[Qui51] W. V. O. Quine. *Mathematical Logic*. Harvard University Press, 1951.

[Smu68] R. M. Smullyan. *First-Order Logic*. Springer-Verlag, 1968.

[Smu94] R. M. Smullyan. *Gödel's Incompleteness Theorems*. Oxford University Press, 1994.