Mathematical Logic

Stephen G. Simpson

October 17, 2013

Department of Mathematics
The Pennsylvania State University
University Park, State College PA 16802

http://www.math.psu.edu/simpson/

This is a set of lecture notes for introductory courses in mathematical logic offered at the Pennsylvania State University.

Contents

Chapter 1

Propositional Calculus

1.1 Formulas

Definition 1.1.1. The *propositional connectives* are *negation* ($\neg$), *conjunction* ($\wedge$), *disjunction* ($\vee$), *implication* ($\Rightarrow$), *biimplication* ($\Leftrightarrow$). They are read as "not", "and", "or", "if-then", "if and only if" respectively. The connectives $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$ are designated as *binary*, while $\neg$ is designated as *unary*.

Definition 1.1.2. A *propositional language* L is a set of *propositional atoms* $p, q, r, \ldots$. An *atomic L-formula* is an atom of L.

Definition 1.1.3. The set of *L-formulas* is generated inductively according to the following rules:

1. If p is an atomic L-formula, then p is an L-formula.

2. If A is an L-formula, then $(\neg A)$ is an L-formula.

3. If A and B are L-formulas, then $(A \wedge B)$, $(A \vee B)$, $(A \Rightarrow B)$, and $(A \Leftrightarrow B)$ are L-formulas.

Note that rule 3 can be written as follows:

3′. If A and B are L-formulas and b is a binary connective, then $(A\,b\,B)$ is an L-formula.

Example 1.1.4. Assume that L contains propositional atoms p, q, r, s. Then

$$(((p \Rightarrow q) \wedge (q \vee r)) \Rightarrow (p \vee r)) \Rightarrow \neg\,(q \vee s)$$

is an L-formula.

Definition 1.1.5. If A is a formula, the *degree* of A is the number of occurrences of propositional connectives in A. This is the same as the number of times rules 2 and 3 had to be applied in order to generate A.

Example 1.1.6. The degree of the formula of Example 1.1.4 is 8.

Remark 1.1.7 (omitting parentheses)**.** As in the above example, we omit parentheses when this can be done without ambiguity. In particular, outermost parentheses can always be omitted, so instead of $((\neg A) \Rightarrow B)$ we may write $(\neg A) \Rightarrow B$. But we may not write $\neg A \Rightarrow B$, because this would not distinguish the intended formula from $\neg(A \Rightarrow B)$.

Definition 1.1.8. Let L be a propositional language. A *formation sequence* is finite sequence $A_1, A_2, \ldots, A_n$ such that each term of the sequence is obtained from previous terms by application of one of the rules in Definition 1.1.3. A *formation sequence for* A is a formation sequence whose last term is A. Note that A is an L-formula if and only if there exists a formation sequence for A.

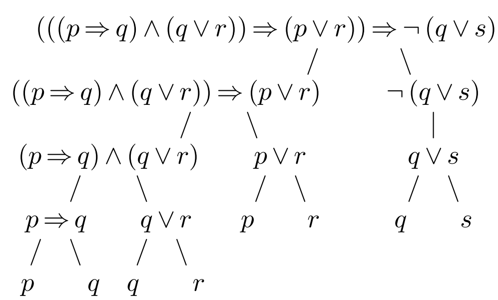
Example 1.1.9. A formation sequence for the L-formula of Example 1.1.4 is

$$p,\ q,\ p \Rightarrow q,\ r,\ q \vee r,\ (p \Rightarrow q) \wedge (q \vee r),\ p \vee r,\ ((p \Rightarrow q) \wedge (q \vee r)) \Rightarrow (p \vee r),$$
$$s,\ q \vee s,\ \neg(q \vee s),\ (((p \Rightarrow q) \wedge (q \vee r)) \Rightarrow (p \vee r)) \Rightarrow \neg(q \vee s)\ .$$

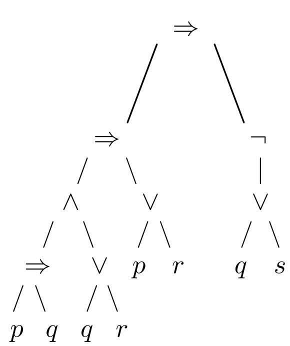
Remark 1.1.10. In contexts where the language L does not need to be specified, an L-formula may be called a *formula*.

Definition 1.1.11. A *formation tree* is a finite rooted dyadic tree where each node carries a formula and each non-atomic formula branches to its immediate subformulas (see the example below). If A is a formula, the *formation tree for* A is the unique formation tree which carries A at its root.

Example 1.1.12. The formation tree for the formula of Example 1.1.4 is



or, in an abbreviated style,



Remark 1.1.13. Note that, if we identify formulas with formation trees in the abbreviated style, then there is no need for parentheses.

Remark 1.1.14. Another way to avoid parentheses is to use Polish notation. In this case the set of L-formulas is generated as follows:

1. If p is an atomic L-formula, then p is an L-formula.

2. If A is an L-formula, then $\neg A$ is an L-formula.

3. If A and B are L-formulas and b is a binary connective, then $b\,A\,B$ is an L-formula.

For example, $(\neg p) \Rightarrow q$ becomes $\Rightarrow \neg p\ q$, and $\neg (p \Rightarrow q)$ becomes $\neg \Rightarrow p\ q$. The formula of Example 1.1.4 becomes

$$\Rightarrow\ \Rightarrow\ \wedge \Rightarrow p\ q\ \vee\ q\ r\ \vee\ p\,r\ \neg\ \vee\ q\ s$$

and a formation sequence for this is

$$p,\ q,\ \Rightarrow p\,q,\ r,\ \vee\, q\,r,\ \wedge \Rightarrow p\,q \vee q\,r,\ \vee\, p\,r,\ \Rightarrow\ \wedge \Rightarrow p\ q \vee q\ r \vee p\ r,$$
$$s,\ \vee\, q\,s,\ \neg\ \vee\, q\,s,\ \Rightarrow\ \Rightarrow\ \wedge \Rightarrow p\ q \vee q\ r \vee p\ r \neg\ \vee q\ s\ .$$

Obviously Polish notation is difficult to read, but it has the advantages of being linear and of not using parentheses.

Remark 1.1.15. In our study of formulas, we shall be indifferent to the question of which system of notation is actually used. The only point of interest for us is that each non-atomic formula is uniquely of the form $\neg A$ or $A\,b\,B$, where A and B are formulas and b is a binary connective.

Exercises 1.1.16. Let C be the formula $(p \wedge \neg q) \Rightarrow \neg (p \vee r)$.

1. Restore all the omitted parentheses to C. (See Remark 1.1.7.)

2. Exhibit a formation sequence for C.

3. List the immediate subformulas of C, their immediate subformulas, etc., i.e., all subformulas of C.

4. Calculate the degrees of C and its subformulas.

5. Display the formation tree of C.

6. Write C according to various notation systems:

   (a) The rules 1–3 of Definition 1.1.3:

      1. Each atom is a formula.
      2. If A is a formula then $(\neg A)$ is a formula.
      3. If A and B are formulas and b is a binary connective, then (AbB) is a formula.

(b) The following alternative set of rules:

   1. Each atom is a formula.
   2. If A is a formula then $\neg(A)$ is a formula.
   3. If A and B are formulas and b is a binary connective, then $(A)b(B)$ is a formula.

(c) Polish notation.

(d) Reverse Polish notation.

1.2 Assignments and Satisfiability

Definition 1.2.1. There are two *truth values*, T and F, denoting truth and falsity.

Definition 1.2.2. Let L be a propositional language. An L-*assignment* is a mapping

$$M : \{p \mid p \text{ is an atomic } L\text{-formula}\} \to \{\mathrm{T}, \mathrm{F}\}\,.$$

Note that if L has exactly n atoms then there are exactly 2^n different L-assignments.

Lemma 1.2.3. Given an L-assignment M, there is a unique L-*valuation*

$$v_M : \{A \mid A \text{ is an } L\text{-formula}\} \to \{\mathrm{T}, \mathrm{F}\}$$

given by the following clauses:

1. $v_M(\neg A) = \begin{cases} \mathrm{T} & \text{if } v_M(A) = \mathrm{F}\,, \\ \mathrm{F} & \text{if } v_M(A) = \mathrm{T}\,. \end{cases}$
2. $v_M(A \wedge B) = \begin{cases} \mathrm{T} & \text{if } v_M(A) = v_M(B) = \mathrm{T}\,, \\ \mathrm{F} & \text{if at least one of } v_M(A), v_M(B) = \mathrm{F}\,. \end{cases}$
3. $v_M(A \vee B) = \begin{cases} \mathrm{T} & \text{if at least one of } v_M(A), v_M(B) = \mathrm{T}\,, \\ \mathrm{F} & \text{if } v_M(A) = v_M(B) = \mathrm{F}\,. \end{cases}$
4. $v_M(A \Rightarrow B) = v_M(\neg(A \wedge \neg B))\,.$
5. $v_M(A \Leftrightarrow B) = \begin{cases} \mathrm{T} & \text{if } v_M(A) = v_M(B)\,, \\ \mathrm{F} & \text{if } v_M(A) \neq v_M(B)\,. \end{cases}$

Proof. The truth value $v_M(A)$ is defined by recursion on L-formulas, i.e., by induction on the degree of A where A is an arbitrary L-formula. □

Remark 1.2.4. Note that each clause of Lemma 1.2.3 corresponds to the familiar truth table for the corresponding propositional connective. Thus clause 3 corresponds to the truth table

A	B	$A \vee B$
T	T	T
T	F	T
F	T	T
F	F	F

for $\vee$, and clause 4 corresponds to the truth table

A	B	$A \Rightarrow B$
T	T	T
T	F	F
F	T	T
F	F	T

for $\Rightarrow$.

Remark 1.2.5. Lemma 1.2.3 may be visualized in terms of formation trees. To define $v_M(A)$ for a formula A, one begins with an assignment of truth values to the atoms, i.e., the end nodes of the formation tree for A, and then proceeds upward to the root, assigning truth values to the nodes, each step being given by the appropriate clause.

Example 1.2.6. Consider the formula $(p \Rightarrow q) \Rightarrow (q \Rightarrow r)$ under an assignment M with $M(p) = \mathrm{T}$, $M(q) = \mathrm{F}$, $M(r) = \mathrm{T}$. In terms of the formation tree, this looks like

$$\begin{array}{ccccccc}
 & & & (p \Rightarrow q) \Rightarrow (q \Rightarrow r) & & & \\
 & & / & & \backslash & & \\
 & p \Rightarrow q & & & & q \Rightarrow r & \\
/ & & \backslash & & / & & \backslash \\
p & & q & & q & & r \\
\mathrm{T} & & \mathrm{F} & & \mathrm{F} & & \mathrm{T}
\end{array}$$

and by applying clause 4 three times we get

$$\begin{array}{ccccccc}
 & & & (p \Rightarrow q) \Rightarrow (q \Rightarrow r) & & & \\
 & & & \mathrm{T} & & & \\
 & & / & & \backslash & & \\
 & p \Rightarrow q & & & & q \Rightarrow r & \\
 & \mathrm{F} & & & & \mathrm{T} & \\
/ & & \backslash & & / & & \backslash \\
p & & q & & q & & r \\
\mathrm{T} & & \mathrm{F} & & \mathrm{F} & & \mathrm{T}
\end{array}$$

and from this we see that $v_M((p \Rightarrow q) \Rightarrow (q \Rightarrow r)) = \mathrm{T}$.

Remark 1.2.7. The above formation tree with truth values can be compressed and written linearly as

$$\begin{array}{l}
(p \Rightarrow q) \Rightarrow (q \Rightarrow r) \\
\mathrm{T\ F\ F\ \ T\ \ F\ T\ T}\,.
\end{array}$$

This illustrates a convenient method for calculating $v_M(A)$, where M is an arbitrary L-assignment.

Remark 1.2.8. Lemma 1.2.3 implies that there is an obvious one-to-one correspondence between L-assignments and L-valuations. If the language L is understood from context, we may speak simply of assignments and valuations.

We now present some key definitions. Fix a propositional language L.

Definition 1.2.9. Let M be an assignment. A formula A is said to be *true under* M if $v_M(A) = \mathrm{T}$, and *false under* M if $v_M(A) = \mathrm{F}$.

Definition 1.2.10. A set of formulas S is said to be *satisfiable* if there exists an assignment M which *satisfies* S, i.e., $v_M(A) = \mathrm{T}$ for all $A \in S$.

Definition 1.2.11. Let S be a set of formulas. A formula B is said to be a *logical consequence of* S if it is true under all assignments which satisfy S.

Definition 1.2.12. A formula B is said to be *logically valid* (or a *tautology*) if B is true under all assignments. Equivalently, B is a logical consequence of the empty set.

Remark 1.2.13. B is a logical consequence of $A_1, \ldots, A_n$ if and only if

$$(A_1 \wedge \cdots \wedge A_n) \Rightarrow B$$

is logically valid. B is logically valid if and only if $\neg B$ is not satisfiable.

Exercises 1.2.14.

1. Use truth tables to show that $((A \Rightarrow B) \Rightarrow A) \Rightarrow A$ is logically valid.

2. Use truth tables to show that $(A \wedge B) \Rightarrow C$ is logically equivalent to $A \Rightarrow (B \Rightarrow C)$.

Exercises 1.2.15. Prove the following. (See Remarks 1.2.13 and 1.3.2.)

1. B is logically valid if and only if $\neg B$ is not satisfiable.

2. B is satisfiable if and only if $\neg B$ is not logically valid.

3. B is a logical consequence of $A_1, \ldots, A_n$ if and only if $(A_1 \wedge \cdots \wedge A_n) \Rightarrow B$ is logically valid.

4. A is logically equivalent to B if and only if $A \Leftrightarrow B$ is logically valid.

Exercise 1.2.16. Brown, Jones, and Smith are suspected of a crime. They testify as follows:

> Brown: Jones is guilty and Smith is innocent.
>
> Jones: If Brown is guilty then so is Smith.

Smith: I'm innocent, but at least one of the others is guilty.

Let b, j, and s be the statements "Brown is innocent," "Jones is innocent," "Smith is innocent".

1. Express the testimony of each suspect as a propositional formula. Write a truth table for the three testimonies.

2. Use the above truth table to answer the following questions:

   (a) Are the three testimonies consistent?

   (b) The testimony of one of the suspects follows from that of another. Which from which?

   (c) Assuming everybody is innocent, who committed perjury?

   (d) Assuming all testimony is true, who is innocent and who is guilty?

   (e) Assuming that the innocent told the truth and the guilty told lies, who is innocent and who is guilty?

Solution.

1. The testimonies are:

$$\begin{array}{ll} B: & (\neg j) \wedge s \\ J: & (\neg b) \Rightarrow (\neg s) \\ S: & s \wedge ((\neg b) \vee (\neg j)) \end{array}$$

   The truth table is:

	b	j	s	B	J	S
1	T	T	T	F	T	F
2	T	T	F	F	T	F
3	T	F	T	T	T	T
4	T	F	F	F	T	F
5	F	T	T	F	F	T
6	F	T	F	F	T	F
7	F	F	T	T	F	T
8	F	F	F	F	T	F

2. (a) Yes, by line 3 of the table.

   (b) The table shows that S is a logical consequence of B. In other words, Smith's testimony follows from Brown's.

   (c) If everybody is innocent, we are in line 1 of the table. Hence B and S are false, i.e., Brown and Smith lied.

   (d) If all the testimony is true, we are in line 3 of the table. Thus Brown and Smith are innocent, while Jones is guilty.

   (e) Our assumption is $v_M(b) = v_M(B)$, $v_M(j) = v_M(J)$, $v_M(s) = v_M(S)$. Hence we are in line 6 of the table. Thus Jones is innocent, and Brown and Smith are guilty.

1.3 Logical Equivalence

Definition 1.3.1. Two formulas A and B are said to be *logically equivalent*, written $A \equiv B$, if each is a logical consequence of the other.

Remark 1.3.2. $A \equiv B$ holds if and only if $A \Leftrightarrow B$ is logically valid.

Exercise 1.3.3. Assume $A_1 \equiv A_2$. Show that

1. $\neg A_1 \equiv \neg A_2$;
2. $A_1 \wedge B \equiv A_2 \wedge B$;
3. $B \wedge A_1 \equiv B \wedge A_2$;
4. $A_1 \vee B \equiv A_2 \vee B$;
5. $B \vee A_1 \equiv B \vee A_2$;
6. $A_1 \Rightarrow B \equiv A_2 \Rightarrow B$;
7. $B \Rightarrow A_1 \equiv B \Rightarrow A_2$;
8. $A_1 \Leftrightarrow B \equiv A_2 \Leftrightarrow B$;
9. $B \Leftrightarrow A_1 \equiv B \Leftrightarrow A_2$.

Exercise 1.3.4. Assume $A_1 \equiv A_2$. Show that for any formula C containing A_1 as a part, if we replace one or more occurrences of the part A_1 by A_2, then the resulting formula is logically equivalent to C. (Hint: Use the results of the previous exercise, plus induction on the degree of C.)

Remark 1.3.5. Some useful logical equivalences are:

1. commutative laws:

   (a) $A \wedge B \equiv B \wedge A$

   (b) $A \vee B \equiv B \vee A$

   (c) $A \Leftrightarrow B \equiv B \Leftrightarrow A$

   Note however that $A \Rightarrow B \not\equiv B \Rightarrow A$.

2. associative laws:

   (a) $A \wedge (B \wedge C) \equiv (A \wedge B) \wedge C$

   (b) $A \vee (B \vee C) \equiv (A \vee B) \vee C$

   (c) $A \Leftrightarrow (B \Leftrightarrow C) \equiv (A \Leftrightarrow B) \Leftrightarrow C$

   Note however that $A \Rightarrow (B \Rightarrow C) \not\equiv (A \Rightarrow B) \Rightarrow C$.

3. distributive laws:

(a) $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$
(b) $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$
(c) $A \Rightarrow (B \wedge C) \equiv (A \Rightarrow B) \wedge (A \Rightarrow C)$
(d) $A \Rightarrow (B \vee C) \equiv (A \Rightarrow B) \vee (A \Rightarrow C)$
(e) $(A \wedge B) \Rightarrow C \equiv (A \Rightarrow C) \vee (B \Rightarrow C)$
(f) $(A \vee B) \Rightarrow C \equiv (A \Rightarrow C) \wedge (B \Rightarrow C)$

Note however that $(A \wedge B) \Rightarrow C \not\equiv (A \Rightarrow C) \wedge (B \Rightarrow C)$, and $(A \vee B) \Rightarrow C \not\equiv (A \Rightarrow C) \vee (B \Rightarrow C)$.

4. negation laws:

(a) $\neg (A \wedge B) \equiv (\neg A) \vee (\neg B)$
(b) $\neg (A \vee B) \equiv (\neg A) \wedge (\neg B)$
(c) $\neg \neg A \equiv A$
(d) $\neg (A \Rightarrow B) \equiv A \wedge \neg B$
(e) $\neg (A \Leftrightarrow B) \equiv (\neg A) \Leftrightarrow B$
(f) $\neg (A \Leftrightarrow B) \equiv A \Leftrightarrow (\neg B)$

5. implication laws:

(a) $A \Rightarrow B \equiv \neg (A \wedge \neg B)$
(b) $A \Rightarrow B \equiv (\neg A) \vee B$
(c) $A \Rightarrow B \equiv (\neg B) \Rightarrow (\neg A)$
(d) $A \Leftrightarrow B \equiv (A \Rightarrow B) \wedge (B \Rightarrow A)$
(e) $A \Leftrightarrow B \equiv (\neg A) \Leftrightarrow (\neg B)$

Definition 1.3.6. A formula is said to be in *disjunctive normal form* if it is of the form $A_1 \vee \cdots \vee A_m$, where each clause A_i, $i = 1, \ldots, m$, is of the form $B_1 \wedge \cdots \wedge B_n$, and each B_j, $j = 1, \ldots, n$ is either an atom or the negation of an atom.

Example 1.3.7. Writing $\overline{p}$ as an abbreviation for $\neg p$, the formula

$$(p_1 \wedge \overline{p_2} \wedge p_3) \vee (\overline{p_1} \wedge p_2 \wedge p_3) \vee (p_1 \wedge \overline{p_2} \wedge \overline{p_3})$$

is in disjunctive normal form.

Exercise 1.3.8. Show that every propositional formula C is logically equivalent to a formula in disjunctive normal form.

Remark 1.3.9. There are two ways to do Exercise 1.3.8.

1. One way is to apply the equivalences of Remark 1.3.5 to subformulas of C via Exercise 1.3.4, much as one applies the commutative and distributive laws in algebra to reduce every algebraic expression to a polynomial.

2. The other way is to use a truth table for C. The disjunctive normal form of C has a clause for each assignment making C true. The clause specifies the assignment.

Example 1.3.10. Consider the formula $(p \Rightarrow q) \Rightarrow r$. We wish to put this in disjunctive normal form.

Method 1. Applying the equivalences of Remark 1.3.5, we obtain

$$\begin{aligned} (p \Rightarrow q) \Rightarrow r &\equiv r \vee \neg (p \Rightarrow q) \\ &\equiv r \vee \neg \neg (p \wedge \neg q) \\ &\equiv r \vee (p \wedge \neg q) \end{aligned}$$

and this is in disjunctive normal form.

Method 2. Consider the truth table

	p	q	r	$p \Rightarrow q$	$(p \Rightarrow q) \Rightarrow r$
1	T	T	T	T	T
2	T	T	F	T	F
3	T	F	T	F	T
4	T	F	F	F	T
5	F	T	T	T	T
6	F	T	F	T	F
7	F	F	T	T	T
8	F	F	F	T	F

Each line of this table corresponds to a different assignment. From lines 1, 3, 4, 5, 7 we read off the following formula equivalent to $(p \Rightarrow q) \Rightarrow r$ in disjunctive normal form:

$$(p \wedge q \wedge r) \vee (p \wedge \overline{q} \wedge r) \vee (p \wedge \overline{q} \wedge \overline{r}) \vee (\overline{p} \wedge q \wedge r) \vee (\overline{p} \wedge \overline{q} \wedge r) \ .$$

1.4 The Tableau Method

Remark 1.4.1. A more descriptive name for tableaux is satisfiability trees. We follow the approach of Smullyan [4].

Definition 1.4.2. A *signed formula* is an expression of the form TA or FA, where A is a formula. An *unsigned formula* is simply a formula.

Definition 1.4.3. A *signed tableau* is a rooted dyadic tree where each node carries a signed formula. An *unsigned tableau* is a rooted dyadic tree where each node carries an unsigned formula. The *signed tableau rules* are presented in Table 1.1. The *unsigned tableau rules* are presented in Table 1.2. If τ is a (signed or unsigned) tableau, an *immediate extension* of τ is a larger tableau τ' obtained by applying a tableau rule to a finite path of τ.

$$\begin{array}{c} \vdots \\ \mathrm{T}\,A \wedge B \\ \vdots \\ | \\ \mathrm{T}\,A \\ \mathrm{T}\,B \end{array} \qquad\qquad \begin{array}{cc} \multicolumn{2}{c}{\vdots} \\ \multicolumn{2}{c}{\mathrm{F}\,A \wedge B} \\ \multicolumn{2}{c}{\vdots} \\ / & \backslash \\ \mathrm{F}\,A & \mathrm{F}\,B \end{array}$$

$$\begin{array}{cc} \multicolumn{2}{c}{\vdots} \\ \multicolumn{2}{c}{\mathrm{T}\,A \vee B} \\ \multicolumn{2}{c}{\vdots} \\ / & \backslash \\ \mathrm{T}\,A & \mathrm{T}\,B \end{array} \qquad\qquad \begin{array}{c} \vdots \\ \mathrm{F}\,A \vee B \\ \vdots \\ | \\ \mathrm{F}\,A \\ \mathrm{F}\,B \end{array}$$

$$\begin{array}{cc} \multicolumn{2}{c}{\vdots} \\ \multicolumn{2}{c}{\mathrm{T}\,A \Rightarrow B} \\ \multicolumn{2}{c}{\vdots} \\ / & \backslash \\ \mathrm{F}\,A & \mathrm{T}\,B \end{array} \qquad\qquad \begin{array}{c} \vdots \\ \mathrm{F}\,A \Rightarrow B \\ \vdots \\ | \\ \mathrm{T}\,A \\ \mathrm{F}\,B \end{array}$$

$$\begin{array}{cc} \multicolumn{2}{c}{\vdots} \\ \multicolumn{2}{c}{\mathrm{T}\,A \Leftrightarrow B} \\ \multicolumn{2}{c}{\vdots} \\ / & \backslash \\ \mathrm{T}\,A & \mathrm{F}\,A \\ \mathrm{T}\,B & \mathrm{F}\,B \end{array} \qquad\qquad \begin{array}{cc} \multicolumn{2}{c}{\vdots} \\ \multicolumn{2}{c}{\mathrm{F}\,A \Leftrightarrow B} \\ \multicolumn{2}{c}{\vdots} \\ / & \backslash \\ \mathrm{T}\,A & \mathrm{F}\,A \\ \mathrm{F}\,B & \mathrm{T}\,B \end{array}$$

$$\begin{array}{c} \vdots \\ \mathrm{T}\,\neg A \\ \vdots \\ | \\ \mathrm{F}\,A \end{array} \qquad\qquad \begin{array}{c} \vdots \\ \mathrm{F}\,\neg A \\ \vdots \\ | \\ \mathrm{T}\,A \end{array}$$

Table 1.1: Signed tableau rules for propositional connectives.

$$\begin{array}{c} \vdots \\ A \wedge B \\ \vdots \\ | \\ A \\ B \end{array} \qquad\qquad \begin{array}{ccc} & \vdots & \\ & \neg\,(A \wedge B) & \\ & \vdots & \\ / & & \backslash \\ \neg\,A & & \neg\,B \end{array}$$

$$\begin{array}{ccc} & \vdots & \\ & A \vee B & \\ & \vdots & \\ / & & \backslash \\ A & & B \end{array} \qquad\qquad \begin{array}{c} \vdots \\ \neg\,(A \vee B) \\ \vdots \\ | \\ \neg\,A \\ \neg\,B \end{array}$$

$$\begin{array}{ccc} & \vdots & \\ & A \Rightarrow B & \\ & \vdots & \\ / & & \backslash \\ \neg\,A & & B \end{array} \qquad\qquad \begin{array}{c} \vdots \\ \neg\,(A \Rightarrow B) \\ \vdots \\ | \\ A \\ \neg\,B \end{array}$$

$$\begin{array}{ccc} & \vdots & \\ & A \Leftrightarrow B & \\ & \vdots & \\ / & & \backslash \\ A & & \neg\,A \\ B & & \neg\,B \end{array} \qquad\qquad \begin{array}{ccc} & \vdots & \\ & \neg\,(A \Leftrightarrow B) & \\ & \vdots & \\ / & & \backslash \\ A & & \neg\,A \\ \neg\,B & & B \end{array}$$

$$\begin{array}{c} \vdots \\ \neg\,\neg\,A \\ \vdots \\ | \\ A \end{array}$$

Table 1.2: Unsigned tableau rules for propositional connectives.

Definition 1.4.4. Let $X_1, \ldots, X_k$ be a finite set of signed or unsigned formulas. A *tableau starting with* $X_1, \ldots, X_k$ is a tableau obtained from

$$\begin{array}{c} X_1 \\ \vdots \\ X_k \end{array}$$

by repeatedly applying tableau rules.

Definition 1.4.5. A path of a tableau is said to be *closed* if it contains a conjugate pair of signed or unsigned formulas, i.e., a pair such as $\mathrm{T}A$, $\mathrm{F}A$ in the signed case, or A, $\neg A$ in the unsigned case. A path of a tableau is said to be *open* if it is not closed. A tableau is said to be *closed* if each of its paths is closed.

The tableau method:

1. To test a formula A for validity, form a signed tableau starting with $\mathrm{F}A$, or equivalently an unsigned tableau starting with $\neg A$. If the tableau closes off, then A is logically valid.

2. To test whether B is a logical consequence of $A_1, \ldots, A_k$, form a signed tableau starting with $\mathrm{T}A_1, \ldots, \mathrm{T}A_k, \mathrm{F}B$, or equivalently an unsigned tableau starting with $A_1, \ldots, A_k, \neg B$. If the tableau closes off, then B is indeed a logical consequence of $A_1, \ldots, A_k$.

3. To test $A_1, \ldots, A_k$ for satisfiability, form a signed tableau starting with $\mathrm{T}A_1, \ldots, \mathrm{T}A_k$, or equivalently an unsigned tableau starting with $A_1, \ldots, A_k$. If the tableau closes off, then $A_1, \ldots, A_k$ is not satisfiable. If the tableau does not close off, then $A_1, \ldots, A_k$ is satisfiable, and from any open path we can read off an assignment satisfying $A_1, \ldots, A_k$.

The correctness of these tests will be proved in Section 1.5. See Corollaries 1.5.9, 1.5.10, 1.5.11 below.

Example 1.4.6. Using the signed tableau method to test $(p \wedge q) \Rightarrow (q \wedge p)$ for logical validity, we have

$$\begin{array}{c} \mathrm{F}\,(p \wedge q) \Rightarrow (q \wedge p) \\ \mathrm{T}\,p \wedge q \\ \mathrm{F}\,q \wedge p \\ \mathrm{T}\,p \\ \mathrm{T}\,q \\ / \quad \backslash \\ \mathrm{F}\,q \qquad \mathrm{F}\,p \end{array}$$

Since (every path of) the tableau is closed, $(p \wedge q) \Rightarrow (q \wedge p)$ is logically valid.

Exercises 1.4.7.

1. Use a signed tableau to show that $(A \Rightarrow B) \Rightarrow (A \Rightarrow C)$ is a logical consequence of $A \Rightarrow (B \Rightarrow C)$.

   Solution.

$$\begin{array}{c} \mathrm{T}\, A \Rightarrow (B \Rightarrow C) \\ \mathrm{F}\, (A \Rightarrow B) \Rightarrow (A \Rightarrow C) \\ \mathrm{T}\, A \Rightarrow B \\ \mathrm{F}\, A \Rightarrow C \\ \mathrm{T}\, A \\ \mathrm{F}\, C \\ / \quad \backslash \\ \begin{array}[t]{cc} \mathrm{F}\, A & \begin{array}[t]{c} \mathrm{T}\, B \Rightarrow C \\ / \qquad \backslash \\ \begin{array}[t]{cc} \begin{array}[t]{c} \mathrm{F}\, B \\ / \quad \backslash \\ \mathrm{F}\, A \quad \mathrm{T}\, B \end{array} & \mathrm{T}\, C \end{array} \end{array} \end{array} \end{array}$$

2. Use a signed tableau to show that $A \Rightarrow B$ is logically equivalent to $(\neg B) \Rightarrow (\neg A)$.

   Solution.

$$\begin{array}{c} \mathrm{F}\, (A \Rightarrow B) \Leftrightarrow ((\neg B) \Rightarrow (\neg A)) \\ / \qquad\qquad \backslash \\ \begin{array}[t]{cc} \begin{array}[t]{c} \mathrm{T}\, A \Rightarrow B \\ \mathrm{F}\, (\neg B) \Rightarrow (\neg A) \\ \mathrm{T}\, \neg B \\ \mathrm{F}\, \neg A \\ \mathrm{F}\, B \\ \mathrm{T}\, A \\ / \quad \backslash \\ \mathrm{F}\, A \quad \mathrm{T}\, B \end{array} & \begin{array}[t]{c} \mathrm{F}\, A \Rightarrow B \\ \mathrm{T}\, (\neg B) \Rightarrow (\neg A) \\ \mathrm{T}\, A \\ \mathrm{F}\, B \\ / \quad \backslash \\ \begin{array}[t]{cc} \begin{array}[t]{c} \mathrm{F}\, \neg B \\ \mathrm{T}\, B \end{array} & \begin{array}[t]{c} \mathrm{T}\, \neg A \\ \mathrm{F}\, A \end{array} \end{array} \end{array} \end{array} \end{array}$$

3. Use an unsigned tableau to show that $A \Rightarrow (B \Rightarrow C)$ is logically equivalent to $(A \wedge B) \Rightarrow C$.

   Solution.

$$\begin{array}{c} \neg\, ((A \Rightarrow (B \Rightarrow C)) \Leftrightarrow ((A \wedge B) \Rightarrow C)) \\ / \qquad\qquad \backslash \\ \begin{array}[t]{cc} \begin{array}[t]{c} A \Rightarrow (B \Rightarrow C) \\ \neg\, ((A \wedge B) \Rightarrow C) \\ A \wedge B \\ \neg\, C \\ A \\ B \\ / \quad \backslash \\ \begin{array}[t]{cc} \neg\, A & \begin{array}[t]{c} B \Rightarrow C \\ / \quad \backslash \\ \neg\, B \quad C \end{array} \end{array} \end{array} & \begin{array}[t]{c} \neg\, (A \Rightarrow (B \Rightarrow C)) \\ (A \wedge B) \Rightarrow C \\ A \\ \neg\, (B \Rightarrow C) \\ B \\ \neg\, C \\ / \quad \backslash \\ \begin{array}[t]{cc} \begin{array}[t]{c} \neg\, (A \wedge B) \\ / \quad \backslash \\ \neg\, A \quad \neg\, B \end{array} & C \end{array} \end{array} \end{array} \end{array}$$

4. Use an unsigned tableau to test $(p \vee q) \Rightarrow (p \wedge q)$ for logical validity. If this formula is not logically valid, use the tableau to find all assignments which falsify it.

   Solution.

$$
\begin{array}{ccccccc}
 & & & \neg\,((p \vee q) \Rightarrow (p \wedge q)) & & & \\
 & & & p \vee q & & & \\
 & & & \neg\,(p \wedge q) & & & \\
 & & / & & \backslash & & \\
 & p & & & & q & \\
 / & & \backslash & & / & & \backslash \\
\neg\,p & & \neg\,q & & \neg\,p & & \neg\,q \\
1 & & 2 & & 3 & & 4
\end{array}
$$

   The open paths 2 and 3 provide the assignments M_2 and M_3 which falsify our formula. $M_2(p) = \mathrm{T}$, $M_2(q) = \mathrm{F}$, $M_3(p) = \mathrm{F}$, $M_3(q) = \mathrm{T}$.

5. Redo the previous problem using a signed tableau.

   Solution.

$$
\begin{array}{ccccccc}
 & & & \mathrm{F}\,(p \vee q) \Rightarrow (p \wedge q) & & & \\
 & & & \mathrm{T}\,p \vee q & & & \\
 & & & \mathrm{F}\,p \wedge q & & & \\
 & & / & & \backslash & & \\
 & \mathrm{T}\,p & & & & \mathrm{T}\,q & \\
 / & & \backslash & & / & & \backslash \\
\mathrm{F}\,p & & \mathrm{F}\,q & & \mathrm{F}\,p & & \mathrm{F}\,q \\
1 & & 2 & & 3 & & 4
\end{array}
$$

   The open paths 2 and 3 provide the assignments M_2 and M_3 which falsify our formula. $M_2(p) = \mathrm{T}$, $M_2(q) = \mathrm{F}$, $M_3(p) = \mathrm{F}$, $M_3(q) = \mathrm{T}$.

Exercise 1.4.8.

1. Formulate the following argument as a propositional formula.

   > If it has snowed, it will be poor driving. If it is poor driving, I will be late unless I start early. Indeed, it has snowed. Therefore, I must start early to avoid being late.

   Solution. Use the following atoms.

   s: it has snowed
   p: it is poor driving
   l: I will be late
   e: I start early

The argument can be translated as follows: $s \Rightarrow p$, $p \Rightarrow (l \vee e)$, s, therefore $(\neg\, l) \Rightarrow e$. Written as a single propositional formula, this becomes:

$$((s \Rightarrow p) \wedge (p \Rightarrow (l \vee e)) \wedge s) \Rightarrow ((\neg\, l) \Rightarrow e).$$

2. Use the tableau method to demonstrate that this formula is logically valid.

Solution.

$$
\begin{array}{c}
\mathrm{F}\,((s \Rightarrow p) \wedge (p \Rightarrow (l \vee e)) \wedge s) \Rightarrow ((\neg\, l) \Rightarrow e) \\
\mathrm{T}\,(s \Rightarrow p) \wedge (p \Rightarrow (l \vee e)) \wedge s \\
\mathrm{F}\,(\neg\, l) \Rightarrow e \\
\mathrm{T}\,s \Rightarrow p \\
\mathrm{T}\,(p \Rightarrow (l \vee e)) \wedge s \\
\mathrm{T}\,p \Rightarrow (l \vee e) \\
\mathrm{T}\,s \\
\mathrm{T}\,\neg\, l \\
\mathrm{F}\,e \\
\mathrm{F}\,l
\end{array}
$$

$$
\begin{array}{cccc}
/ & \backslash & & \\
\mathrm{F}\,s & \mathrm{T}\,p & & \\
 & / & \backslash & \\
 & \mathrm{F}\,p & \mathrm{T}\,l \vee e & \\
 & & / & \backslash \\
 & & \mathrm{T}\,l & \mathrm{T}\,e
\end{array}
$$

1.5 The Completeness Theorem

Let $X_1, \ldots, X_k$ be a finite set of signed formulas, or a finite set of unsigned formulas.

Lemma 1.5.1 (the Soundness Theorem)**.** If τ is a finite closed tableau starting with $X_1, \ldots, X_k$, then $X_1, \ldots, X_k$ is not satisfiable.

Proof. Straightforward. □

Definition 1.5.2. A path of a tableau is said to be *replete* if, whenever it contains the top formula of a tableau rule, it also contains at least one of the branches. A *replete tableau* is a tableau in which every path is replete.

Lemma 1.5.3. Any finite tableau can be extended to a finite replete tableau.

Proof. Apply tableau rules until they cannot be applied any more. □

Definition 1.5.4. A tableau is said to be *open* if it is not closed, i.e., it has at least one open path.

Lemma 1.5.5. Let τ be a replete tableau starting with $X_1, \ldots, X_k$. If τ is open, then $X_1, \ldots, X_k$ is satisfiable.

In order to prove Lemma 1.5.5, we introduce the following definition.

Definition 1.5.6. Let S be a set of signed or unsigned formulas. We say that S is a *Hintikka set* if

1. S "obeys the tableau rules", in the sense that if it contains the top formula of a rule then it contains at least one of the branches;

2. S contains no pair of conjugate atomic formulas, i.e., Tp, Fp in the signed case, or $p, \neg p$ in the unsigned case.

Lemma 1.5.7 (Hintikka's Lemma)**.** If S is a Hintikka set, then S is satisfiable.

Proof. Define an assignment M by

$$M(p) \;=\; \begin{cases} \mathrm{T} & \text{if } \mathrm{T}p \text{ belongs to } S \\ \mathrm{F} & \text{otherwise} \end{cases}$$

in the signed case, or

$$M(p) \;=\; \begin{cases} \mathrm{T} & \text{if } p \text{ belongs to } S \\ \mathrm{F} & \text{otherwise} \end{cases}$$

in the unsigned case. It is not difficult to see that $v_M(X) = \mathrm{T}$ for all $X \in S$. $\square$

To prove Lemma 1.5.5, it suffices to note that a replete open path is a Hintikka set. Thus, if a replete tableau starting with $X_1, \ldots, X_k$ is open, Hintikka's Lemma implies that $X_1, \ldots, X_k$ is satisfiable.

Combining Lemmas 1.5.1 and 1.5.3 and 1.5.5, we obtain:

Theorem 1.5.8 (the Completeness Theorem)**.** $X_1, \ldots, X_k$ is satisfiable if and only if there is no finite closed tableau starting with $X_1, \ldots, X_k$.

Corollary 1.5.9. $A_1, \ldots, A_k$ is not satisfiable if and only if there exists a finite closed signed tableau starting with TA_1, …, TA_k, or equivalently a finite closed unsigned tableau starting with $A_1, \ldots, A_k$.

Corollary 1.5.10. A is logically valid if and only if there exists a finite closed signed tableau starting with FA, or equivalently a finite closed unsigned tableau starting with $\neg A$.

Corollary 1.5.11. B is a logical consequence of $A_1, \ldots, A_k$ if and only if there exists a finite closed signed tableau starting with TA_1, …, TA_k, FB, or equivalently a finite closed unsigned tableau starting with $A_1, \ldots, A_k, \neg B$.

Exercise 1.5.12. Consider the following argument.

> The attack will succeed only if the enemy is taken by surprise or the position is weakly defended. The enemy will not be taken by surprise unless he is overconfident. The enemy will not be overconfident if the position is weakly defended. Therefore, the attack will not succeed.

1. Translate the argument into propositional calculus.

2. Use an unsigned tableau to determine whether the argument is logically valid.

1.6 Trees and König's Lemma

Up to this point, our discussion of trees has been informal. We now pause to make our tree terminology precise.

Definition 1.6.1. A *tree* consists of

1. a set T
2. a function $\ell : T \to \mathbb{N}^+$,
3. a binary relation P on T.

The elements of T are called the *nodes* of the tree. For $x \in T$, $\ell(x)$ is the *level* of x. The relation xPy is read as x *immediately precedes* y, or y *immediately succeeds* x. We require that there is exactly one node $x \in T$ such that $\ell(x) = 1$, called the *root* of the tree. We require that each node other than the root has exactly one immediate predecessor. We require that $\ell(y) = \ell(x) + 1$ for all $x, y \in T$ such that xPy.

Definition 1.6.2. A *subtree of* T is a nonempty set $T' \subseteq T$ such that for all $y \in T'$ and xPy, $x \in T'$. Note that T' is itself a tree, under the restriction of ℓ and P to T'. Moreover, the root of T' is the same as the root of T.

Definition 1.6.3. An *end node* of T is a node with no (immediate) successors. A *path* in T is a set $S \subseteq T$ such that (1) the root of T belongs to S, (2) for each $x \in S$, either x is an end node of T or there is exactly one $y \in S$ such that xPy.

Definition 1.6.4. Let P^* be the transitive closure of P, i.e., the smallest reflexive and transitive relation on T containing P. For $x, y \in T$, we have xP^*y if and only if x *precedes* y, i.e., y *succeeds* x, i.e., there exists a finite sequence $x = x_0Px_1Px_2 \cdots x_{n-1}Px_n = y$. Note that the relation P^* is reflexive (xP^*x for all $x \in T$), antisymmetric (xP^*y and yP^*x imply $x = y$), and transitive (xP^*y and yP^*z imply xP^*z). Thus P^* is a partial ordering of T.

Definition 1.6.5. T is *finitely branching* if each node of T has only finitely many immediate successors in T. T is *dyadic* if each node of T has at most two immediate successors in T. Note that a dyadic tree is finitely branching.

Theorem 1.6.6 (König's Lemma)**.** Let T be an infinite, finitely branching tree. Then T has an infinite path.

Proof. Let $\widehat{T}$ be the set of all $x \in T$ such that x has infinitely many successors in T. Note that $\widehat{T}$ is a subtree of T. Since T is finitely branching, it follows by the pigeonhole principle that each $x \in \widehat{T}$ has at least one immediate successor $y \in \widehat{T}$. Now define an infinite path $S = \{x_1, x_2, \ldots, x_n, \ldots\}$ in $\widehat{T}$ inductively by putting $x_1 =$ the root of T, and $x_{n+1} =$ one of the immediate successors of x_n in $\widehat{T}$. Clearly S is an infinite path of T. $\square$

1.7 The Compactness Theorem

Theorem 1.7.1 (the Compactness Theorem, countable case)**.** Let S be a countable set of propositional formulas. If each finite subset of S is satisfiable, then S is satisfiable.

Proof. In brief outline: Form an infinite tableau. Apply König's Lemma to get an infinite path. Apply Hintikka's Lemma.

Details: Let $S = \{A_1, A_2, \ldots, A_i, \ldots\}$. Start with A_1 and generate a finite replete tableau, τ_1. Since A_1 is satisfiable, τ_1 has at least one open path. Append A_2 to each of the open paths of τ_1, and generate a finite replete tableau, τ_2. Since $\{A_1, A_2\}$ is satisfiable, τ_2 has at least one open path. Append A_3 to each of the open paths of τ_2, and generate a finite replete tableau, τ_3. .... Put $\tau = \bigcup_{i=1}^{\infty} \tau_i$. Thus τ is a replete tableau. Note also that τ is an infinite, finitely branching tree. By König's Lemma (Theorem 1.6.6), let S' be an infinite path in τ. Then S' is a Hintikka set containing S. By Hintikka's Lemma, S' is satisfiable. Hence S is satisfiable. $\square$

Theorem 1.7.2 (the Compactness Theorem, uncountable case)**.** Let S be an uncountable set of propositional formulas. If each finite subset of S is satisfiable, then S is satisfiable.

Proof. We present three proofs. The first uses Zorn's Lemma. The second uses transfinite induction. The third uses Tychonoff's Theorem.

Let L be the (necessarily uncountable) propositional language consisting of all atoms occurring in formulas of S. If S is a set of L-formulas, we say that S is *finitely satisfiable* if each finite subset of S is satisfiable. We are trying to prove that, if S is finitely satisfiable, then S is satisfiable.

First proof. Consider the partial ordering $\mathfrak{F}$ of all finitely satisfiable sets of L-formulas which include S, ordered by inclusion. It is easy to see that any chain in $\mathfrak{F}$ has a least upper bound in $\mathfrak{F}$. Hence, by Zorn's Lemma, $\mathfrak{F}$ has a maximal element, S^*. Thus S^* is a set of L-formulas, $S^* \supseteq S$, S^* is finitely satisfiable, and for each L-formula $A \notin S^*$, $S^* \cup \{A\}$ is not finitely satisfiable. From this it is straightforward to verify that S^* is a Hintikka set. Hence, by Hintikka's Lemma, S^* is satisfiable. Hence S is satisfiable.

Second proof. Let A_ξ, $\xi < \alpha$, be a transfinite enumeration of all L-formulas. By transfinite recursion, put $S_0 = S$, $S_{\xi+1} = S_\xi \cup \{A_\xi\}$ if $S_\xi \cup \{A_\xi\}$ is finitely satisfiable, $S_{\xi+1} = S_\xi$ otherwise, and $S_\eta = \bigcup_{\xi<\eta} S_\xi$ for limit ordinals $\eta \leq \alpha$. Using transfinite induction, it is easy to verify that S_ξ is finitely satisfiable for each $\xi \leq \alpha$. In particular, S_α is finitely satisfiable. It is straightforward to verify that S_α is a Hintikka set. Hence, by Hintikka's Lemma, S_α is satisfiable. Hence S is satisfiable.

Third proof. Let $\mathfrak{M} = \{\mathrm{T}, \mathrm{F}\}^L$ be the space of all L-assignments $M : L \to \{\mathrm{T}, \mathrm{F}\}$. Make $\mathfrak{M}$ a topological space with the product topology where $\{\mathrm{T}, \mathrm{F}\}$ has the discrete topology. Since $\{\mathrm{T}, \mathrm{F}\}$ is compact, it follows by Tychonoff's Theorem that $\mathfrak{M}$ is compact. For each L-formula A, put $\mathfrak{M}_A = \{M \in \mathfrak{M} \mid v_M(A) = \mathrm{T}\}$. It is easy to check that each $\mathfrak{M}_A$ is a topologically closed set in $\mathfrak{M}$. If S is finitely satisfiable, then the family of sets $\mathfrak{M}_A$, $A \in S$ has the finite intersection property, i.e., $\bigcap_{A \in S_0} \mathfrak{M}_A \neq \emptyset$ for each finite $S_0 \subseteq S$. By compactness of $\mathfrak{M}$ it follows that $\bigcap_{A \in S} \mathfrak{M}_A \neq \emptyset$. Thus S is satisfiable. $\square$

1.8 Combinatorial Applications

In this section we present some combinatorial applications of the Compactness Theorem for propositional calculus.

Definition 1.8.1.

1. A *graph* consists of a set of *vertices* together with a specification of certain pairs of vertices as being *adjacent*. We require that a vertex may not be adjacent to itself, and that u is adjacent to v if and only if v is adjacent to u.

2. Let G be a graph and let k be a positive integer. A k-*coloring* of G is a function $f : \{\text{vertices of } G\} \to \{c_1, \ldots, c_k\}$ such that $f(u) \neq f(v)$ for all adjacent pairs of vertices u, v.

3. G is said to be k-*colorable* if there exists a k-coloring of G. This notion is much studied in graph theory.

Exercise 1.8.2. Let G be a graph and let k be a positive integer. For each vertex v and each $i = 1, \ldots, k$, let p_{vi} be a propositional atom expressing that vertex v receives color c_i. Define $C_k(G)$ to be the following set of propositional formulas: $p_{v1} \lor \cdots \lor p_{vk}$ for each vertex v; $\neg\,(p_{vi} \land p_{vj})$ for each vertex v and $1 \leq i < j \leq k$; $\neg\,(p_{ui} \land p_{vi})$ for each adjacent pair of vertices u, v and $1 \leq i \leq k$.

1. Show that there is a one-to-one correspondence between k-colorings of G and assignments satisfying $C_k(G)$.

2. Show that G is k-colorable if and only if $C_k(G)$ is satisfiable.

3. Show that G is k-colorable if and only if each finite subgraph of G is k-colorable.

Definition 1.8.3. A *partial ordering* consists of a set P together with a binary relation $\leq_P$ such that

1. $a \leq_P a$ for all $a \in P$ (reflexivity);

2. $a \leq_P b$, $b \leq_P c$ imply $a \leq_P c$ (transitivity);

3. $a \leq_P b$, $b \leq_P a$ imply $a = b$ (antisymmetry).

Example 1.8.4. Let $P = \mathbb{N}^+ = \{1, 2, 3, \ldots, n, \ldots\}$ = the set of positive integers.

1. Let $\leq_P$ be the usual order relation on P, i.e., $m \leq_P n$ if and only if $m \leq n$.

2. Let $\leq_P$ be the divisibility ordering of P, i.e., $m \leq_P n$ if and only if m is a divisor of n.

Definition 1.8.5. Let $P, \leq_P$ be a partial ordering.

1. Two elements $a, b \in P$ are *comparable* if either $a \leq_P b$ or $b \leq_P a$. Otherwise they are *incomparable*.

2. A *chain* is a set $X \subseteq P$ such that any two elements of X are comparable.

3. An *antichain* is a set $X \subseteq P$ such that any two distinct elements of X are incomparable.

Exercise 1.8.6. Let $P, \leq_P$ be a partial ordering, and let k be a positive integer.

1. Use the Compactness Theorem to show that P is the union of k chains if and only if each finite subset of P is the union of k chains.

2. *Dilworth's Theorem* says that P is the union of k chains if and only if every antichain is of size $\leq k$. Show that Dilworth's Theorem for arbitrary partial orderings follows from Dilworth's Theorem for finite partial orderings.

Chapter 2

Predicate Calculus

2.1 Formulas and Sentences

Definition 2.1.1 (languages)**.** A *language* L is a set of *predicates*, each predicate P of L being designated as *n-ary* for some nonnegative[1] integer n.

Definition 2.1.2 (variables and quantifiers)**.** We assume the existence of a fixed, countably infinite set of symbols $x, y, z, \ldots$ known as *variables*. We introduce two new symbols: the *universal quantifier* ($\forall$) and the *existential quantifier* ($\exists$). They are read as "for all" and "there exists", respectively.

Definition 2.1.3 (formulas)**.** Let L be a language, and let U be a set. It is understood that U is disjoint from the set of variables. The set of *L-U-formulas* is generated as follows.

1. An *atomic* L-U-formula is an expression of the form $Pe_1 \cdots e_n$ where P is an n-ary predicate of L and each of $e_1, \ldots, e_n$ is either a variable or an element of U.

2. Each atomic L-U-formula is an L-U-formula.

3. If A is an L-U-formula, then $\neg A$ is an L-U-formula.

4. If A and B are L-U-formulas, then $A \wedge B$, $A \vee B$, $A \Rightarrow B$, $A \Leftrightarrow B$ are L-U-formulas.

5. If x is a variable and A is an L-U-formula, then $\forall x\, A$ and $\exists x\, A$ are L-U-formulas.

Definition 2.1.4 (degree)**.** The *degree* of a formula is the number of occurrences of propositional connectives $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$ and quantifiers $\forall, \exists$ in it.

[1] It will be seen that 0-ary predicates behave as propositional atoms. Thus the predicate calculus is an extension of the propositional calculus.

Definition 2.1.5. An L-*formula* is an L-$\emptyset$-formula, i.e., an L-U-formula where $U = \emptyset$, the empty set.

Remark 2.1.6. If U is a subset of U', then every L-U-formula is automatically an L-U'-formula. In particular, every L-formula is automatically an L-U-formula, for any set U.

Definition 2.1.7. In situations where the language L is understood from context, an L-U-formula may be called a U-*formula*, and an L-formula a *formula*.

Definition 2.1.8 (substitution)**.** If A is an L-U-formula and x is a variable and $a \in U$, we define an L-U-formula $A[x/a]$ as follows.

1. If A is atomic, then $A[x/a] =$ the result of replacing each occurrence of x in A by a.
2. $(\neg A)[x/a] = \neg A[x/a]$.
3. $(A \wedge B)[x/a] = A[x/a] \wedge B[x/a]$.
4. $(A \vee B)[x/a] = A[x/a] \vee B[x/a]$.
5. $(A \Rightarrow B)[x/a] = A[x/a] \Rightarrow B[x/a]$.
6. $(A \Leftrightarrow B)[x/a] = A[x/a] \Leftrightarrow B[x/a]$.
7. $(\forall x\, A)[x/a] = \forall x\, A$.
8. $(\exists x\, A)[x/a] = \exists x\, A$.
9. If y is a variable other than x, then $(\forall y\, A)[x/a] = \forall y\, A[x/a]$.
10. If y is a variable other than x, then $(\exists y\, A)[x/a] = \exists y\, A[x/a]$.

Definition 2.1.9 (free variables)**.** An occurrence of a variable x in an L-U-formula A is said to be *bound in* A if it is within the scope of a quantifier $\forall x$ or $\exists x$ in A. An occurrence of a variable x in an L-U-formula A is said to be *free in* A if it is not bound in A. A variable x is said to *occur freely in* A if there is at least one occurrence of x in A which is free in A.

Exercise 2.1.10.

1. Show that $A[x/a]$ is the result of substituting a for all free occurrences of x in A.
2. Show that x occurs freely in A if and only if $A[x/a] \neq A$.

Definition 2.1.11 (sentences)**.** An L-U-*sentence* is an L-U-formula in which no variables occur freely. An L-*sentence* is an L-$\emptyset$-sentence, i.e., an L-U-sentence where $U = \emptyset$, the empty set.

Remark 2.1.12. If U is a subset of U', then every L-U-sentence is automatically an L-U'-sentence. In particular, every L-sentence is automatically an L-U-sentence, for any set U.

Definition 2.1.13. In situations where the language L is understood from context, an L-U-sentence may be called a U-*sentence*, and an L-sentence a *sentence*.

2.2 Structures and Satisfiability

Definition 2.2.1. Let U be a nonempty set, and let n be a nonnegative[2] integer. U^n is the set of all n-tuples of elements of U, i.e.,

$$U^n = \{\langle a_1, \ldots, a_n \rangle \mid a_1, \ldots, a_n \in U\} .$$

An n-*ary relation on* U is a subset of U^n.

Definition 2.2.2. Let L be a language. An L-*structure* M consists of a nonempty set U_M, called the *domain* or *universe* of M, together with an n-ary relation P_M on U_M for each n-ary predicate P of L. An L-structure may be called a *structure*, in situations where the language L is understood from context.

Definition 2.2.3. Two L-structures M and M' are said to be *isomorphic* if there exists an *isomorphism* of M onto M', i.e., a one-to-one correspondence $\phi : U_M \cong U_{M'}$ such that for all n-ary predicates P of L and all n-tuples $\langle a_1, \ldots, a_n \rangle \in (U_M)^n$, $\langle a_1, \ldots, a_n \rangle \in P_M$ if and only if $\langle \phi(a_1), \ldots, \phi(a_n) \rangle \in P_{M'}$.

As usual in abstract mathematics, we are mainly interested in properties of structures that are invariant under isomorphism.

Lemma 2.2.4. Given an L-structure M, there is a unique *valuation* or assignment of truth values

$$v_M : \{A \mid A \text{ is an } L\text{-}U_M\text{-sentence}\} \to \{\mathrm{T}, \mathrm{F}\}$$

defined as follows:

1. $v_M(Pa_1 \cdots a_n) = \begin{cases} \mathrm{T} & \text{if } \langle a_1, \ldots, a_n \rangle \in P_M \,, \\ \mathrm{F} & \text{if } \langle a_1, \ldots, a_n \rangle \notin P_M \,. \end{cases}$

2. $v_M(\neg A) = \begin{cases} \mathrm{T} & \text{if } v_M(A) = \mathrm{F} \,, \\ \mathrm{F} & \text{if } v_M(A) = \mathrm{T} \,. \end{cases}$

3. $v_M(A \wedge B) = \begin{cases} \mathrm{T} & \text{if } v_M(A) = v_M(B) = \mathrm{T} \,, \\ \mathrm{F} & \text{if at least one of } v_M(A), v_M(B) = \mathrm{F} \,. \end{cases}$

[2] In the special case $n = 0$ we obtain the notion of a 0-ary relation, i.e., a subset of $\{\langle\rangle\}$. There are only two 0-ary relations, $\{\langle\rangle\}$ and $\{\}$, corresponding to T and F respectively. Thus a 0-ary predicate behaves as a propositional atom.

4. $v_M(A \vee B) = \begin{cases} \mathrm{T} & \text{if at least one of } v_M(A), v_M(B) = \mathrm{T}\,, \\ \mathrm{F} & \text{if } v_M(A) = v_M(B) = \mathrm{F}\,. \end{cases}$

5. $v_M(A \Rightarrow B) = v_M(\neg\,(A \wedge \neg\, B))\,.$

6. $v_M(A \Leftrightarrow B) = \begin{cases} \mathrm{T} & \text{if } v_M(A) = v_M(B)\,, \\ \mathrm{F} & \text{if } v_M(A) \neq v_M(B)\,. \end{cases}$

7. $v_M(\forall x\, A) = \begin{cases} \mathrm{T} & \text{if } v_M(A[x/a]) = \mathrm{T} \text{ for all } a \in U_M\,, \\ \mathrm{F} & \text{if } v_M(A[x/a]) = \mathrm{F} \text{ for at least one } a \in U_M\,. \end{cases}$

8. $v_M(\exists x\, A) = \begin{cases} \mathrm{T} & \text{if } v_M(A[x/a]) = \mathrm{T} \text{ for at least one } a \in U_M\,, \\ \mathrm{F} & \text{if } v_M(A[x/a]) = \mathrm{F} \text{ for all } a \in U_M\,. \end{cases}$

Proof. The truth value $v_M(A)$ is defined by recursion on L-U_M-sentences, i.e., by induction on the degree of A where A is an arbitrary L-U_M-sentence. □

Definition 2.2.5 (truth and satisfaction)**.** Let M be an L-structure.

1. Let A be an L-U_M-sentence. We say that A is *true in* M if $v_M(A) = \mathrm{T}$. We say that A is *false in* M if $v_M(A) = \mathrm{F}$.

2. Let S be a set of L-U_M-sentences. We say that M *satisfies* S, abbreviated $M \models S$, if all of the sentences of S are true in M.

Theorem 2.2.6.

1. If M and M' are isomorphic L-structures and $\phi : M \cong M'$ is an isomorphism of M onto M', then for all L-U_M-sentences A we have $v_M(A) = v_{M'}(A')$ where $A' = A[a_1/\phi(a_1), \ldots, a_k/\phi(a_k)]$.[3] Here $a_1, \ldots, a_k$ are the elements of U_M which occur in A.

2. If M and M' are isomorphic L-structures, then they are *elementarily equivalent*, i.e., they satisfy the same L-sentences. We shall see later that the converse does not hold in general.

Proof. We omit the proof of part 1. A more general result will be proved later as Theorem 2.7.3. Part 2 follows immediately from part 1. □

Definition 2.2.7 (satisfiability)**.** Let S be a set of L-sentences. S is said to be *satisfiable*[4] if there exists an L-structure M which satisfies S.

[3] We have extended the substitution notation 2.1.8 in an obvious way.

[4] Similarly, the notions of logical validity and logical consequence are defined for L-sentences, in the obvious way, using L-structures. An L-sentence is said to be *logically valid* if it is satisfied by all L-structures. An L-sentence is said to be a *logical consequence* of S if it is satisfied by all L-structures satisfying S.

Remark 2.2.8. Satisfiability is one of the most important concepts of mathematical logic. A key result known as the Compactness Theorem[5] states that a set S of L-sentences is satisfiable if and only every finite subset of S is satisfiable.

The following related notion is of technical importance.

Definition 2.2.9 (satisfiability in a domain)**.** Let U be a nonempty set. A set S of L-U-sentences is said to be *satisfiable in the domain* U if there exists an L-structure M such that $M \models S$ and $U_M = U$.

Remark 2.2.10. Let S be a set of L-sentences. Then S is satisfiable (according to Definition 2.2.7) if and only if S is satisfiable in some domain U.

Theorem 2.2.11. Let S be a set of L-sentences. If S is satisfiable in a domain U, then S is satisfiable in any domain of the same cardinality as U.

Proof. Suppose S is satisfiable in a domain U. Let M be an L-structure M satisfying S with $U_M = U$. Let U' be any set of the same cardinality as U. Then there exists a one-to-one correspondence $\phi : U \to U'$. Let M' be the L-structure with $U_{M'} = U'$, $P_{M'} = \{\langle \phi(a_1), \ldots, \phi(a_n)\rangle \mid \langle a_1, \ldots, a_n\rangle \in P_M\}$ for all n-ary predicates P of L. Then M is isomorphic to M'. Hence, by Theorem 2.2.6, $M' \models S$. Thus S is satisfiable in the domain U'. $\square$

Example 2.2.12. We exhibit a sentence A_∞ which is satisfiable in an infinite domain but not in any finite domain. Our sentence A_∞ is $(1) \wedge (2) \wedge (3)$ with

$$\begin{array}{ll} (1) & \forall x\, \forall y\, \forall z\, ((Rxy \wedge Ryz) \Rightarrow Rxz) \\ (2) & \forall x\, \forall y\, (Rxy \Rightarrow \neg\, Ryx) \\ (3) & \forall x\, \exists y\, Rxy \end{array}$$

See also Example 2.5.9.

Exercise 2.2.13. Let L be the language consisting of one binary predicate, R. Consider the following sentences:

(a) $\forall x\, Rxx$

(b) $\forall x\, \neg\, Rxx$

(c) $\forall x\, \forall y\, (Rxy \Rightarrow Ryx)$

(d) $\forall x\, \forall y\, (Rxy \Rightarrow \neg\, Ryx)$

(e) $\forall x\, \forall y\, \forall z\, ((Rxy \wedge Ryz) \Rightarrow Rxz)$

(f) $\forall x\, \exists y\, Rxy$

Which of subsets of this set of sentences are satisfiable? Verify your claims by exhibiting appropriate structures. Use the simplest possible structures.

[5] See Theorems 2.6.1 and 2.6.2 below.

Solution.

(a,c,e,f) is satisfiable: $U = \{1\}$, $R = \{\langle 1, 1\rangle\}$.

(b,c,d,e) is satisfiable: $U = \{1\}$, $R = \{\,\}$.

(b,c,f) is satisfiable: $U = \{1, 2\}$, $R = \{\langle 1, 2\rangle, \langle 2, 1\rangle\}$.

(b,d,e,f) is satisfiable: $U = \{1, 2, 3, \ldots\}$, $R = <$.

These are the only maximal satisfiable sets, because:

(a,b) is not satisfiable.

(a,d) is not satisfiable.

(b,c,e,f) is not satisfiable.

(c,d,f) is not satisfiable.

Note: (d,e,f) is not satisfiable in any finite domain.

Exercise 2.2.14.

1. Assume the following predicates:

   Hx: x is a human
   Cx: x is a car
   Tx: x is a truck
   Dxy: x drives y

   Write formulas representing the obvious assumptions: no human is a car, no car is a truck, humans exist, cars exist, only humans drive, only cars and trucks are driven, etc.

2. Write formulas representing the following statements:

   (a) Everybody drives a car or a truck.
   (b) Some people drive both.
   (c) Some people don't drive either.
   (d) Nobody drives both.

3. Assume in addition the following predicate:

   Ixy: x is identical to y

   Write formulas representing the following statements:

   (a) Every car has at most one driver.
   (b) Every truck has exactly two drivers.

(c) Everybody drives exactly one vehicle (car or truck).

Solution.

1. No human is a car. $\neg \exists x\,(Hx \wedge Cx)$.

   No car is a truck. $\neg \exists x\,(Cx \wedge Tx)$.

   Humans exist. $\exists x\, Hx$.

   Cars exist. $\exists x\, Cx$.

   Only humans drive. $\forall x\,((\exists y\, Dxy) \Rightarrow Hx)$.

   Only cars and trucks are driven. $\forall x\,((\exists y\, Dyx) \Rightarrow (Cx \vee Tx))$.

   Some humans drive. $\exists x\,(Hx \wedge \exists y\, Dxy)$.

   Some humans do not drive. $\exists x\,(Hx \wedge \neg \exists y\, Dxy)$.

   Some cars are driven. $\exists x\,(Cx \wedge \exists y\, Dyx)$.

   Some cars are not driven (e.g., old wrecks). $\exists x\,(Cx \wedge \neg \exists y\, Dyx)$.

   Etc, etc.

2. (a) $\forall x\,(Hx \Rightarrow \exists y\,(Dxy \wedge (Cy \vee Ty)))$.
   (b) $\exists x\,(Hx \wedge \exists y\, \exists z\,(Dxy \wedge Cy \wedge Dxz \wedge Tz))$.
   (c) $\exists x\,(Hx \wedge \neg \exists y\,(Dxy \wedge (Cy \vee Ty)))$.
   (d) $\neg \exists x\,(Hx \wedge \exists y\, \exists z\,(Dxy \wedge Dxz \wedge Cy \wedge Tz))$.

3. (a) $\forall x\,(Cx \Rightarrow \forall y\, \forall z\,((Dyx \wedge Dzx) \Rightarrow Iyz))$.
   (b) $\forall x\,(Tx \Rightarrow \exists y\, \exists z\,((\neg Iyz) \wedge Dyx \wedge Dzx \wedge \forall w\,(Dwx \Rightarrow (Iwy \vee Iwz))))$.
   (c) $\forall x\,(Hx \Rightarrow \exists y\,(Dxy \wedge (Cy \vee Ty) \wedge \forall z\,((Dxz \wedge (Cz \vee Tz)) \Rightarrow Iyz)))$.

Exercise 2.2.15. Assume the following predicates:

Ixy: $x = y$

$Pxyz$: $x \cdot y = z$

Write formulas representing the axioms for a group: axioms for equality, existence and uniqueness of products, associative law, existence of an identity element, existence of inverses.

Solution.

1. equality axioms:
   (a) $\forall x\, Ixx$ (reflexivity)
   (b) $\forall x\, \forall y\,(Ixy \Leftrightarrow Iyx)$ (symmetry)
   (c) $\forall x\, \forall y\, \forall z\,((Ixy \wedge Iyz) \Rightarrow Ixz)$ (transitivity)
   (d) $\forall x\, \forall x'\, \forall y\, \forall y'\, \forall z\, \forall z'\,((Ixx' \wedge Iyy' \wedge Izz') \Rightarrow (Pxyz \Leftrightarrow Px'y'z'))$
   (congruence with respect to P).

2. existence and uniqueness of products:

   (a) $\forall x\, \forall y\, \exists z\, Pxyz$ (existence)

   (b) $\forall x\, \forall y\, \forall z\, \forall w\, ((Pxyz \wedge Pxyw) \Rightarrow Izw)$ (uniqueness).

3. associative law:

   $\forall x\, \forall y\, \forall z\, \exists u\, \exists v\, \exists w\, (Pxyu \wedge Pyzv \wedge Puzw \wedge Pxvw)$.

4. existence of identity element:

   $\exists u\, \forall x\, (Puxx \wedge Pxux)$.

5. existence of inverses:

   $\exists u\, \forall x\, \exists y\, (Puxx \wedge Pxux \wedge Pxyu \wedge Pyxu)$.

Exercise 2.2.16. Let G be a group. The *order* of an element $a \in G$ is the smallest positive integer n such that $a^n = e$. Here e denotes the identity element of G, and

$$a^n = \underbrace{a \cdot \cdots \cdot a}_{n \text{ times}}.$$

Using only the predicates $Pxyz$ ("$x \cdot y = z$") and Ixy ("$x = y$"), write a predicate calculus sentence S such that, for any group G, G satisfies S if and only if G has no elements of order 2 or 3.

Solution. $\neg\, \exists x\, \exists y\, (Pxxx \wedge (\neg\, Pyyy) \wedge (Pyyx \vee \exists z\, (Pyyz \wedge Pzyx)))$.

2.3 The Tableau Method

Definition 2.3.1. Fix a countably infinite set $V = \{a_1, a_2, \ldots, a_n, \ldots\} = \{a, b, c, \ldots\}$. The elements of V will be called *parameters*. If L is a language, L-V-sentences will be called *sentences with parameters*.

Definition 2.3.2. A (*signed* or *unsigned*) *tableau* is a rooted dyadic tree where each node carries a (signed or unsigned) L-V-sentence. The tableau rules for the predicate calculus are the same as those for the propositional calculus, with the following additional rules.

Signed:

$$\begin{array}{ccc} \vdots & \qquad\qquad & \vdots \\ \mathrm{T}\, \forall x\, A & & \mathrm{F}\, \exists x\, A \\ \vdots & & \vdots \\ | & & | \\ \mathrm{T}\, A[x/a] & & \mathrm{F}\, A[x/a] \end{array}$$

where a is an arbitrary parameter

$$
\begin{array}{ccc}
\vdots & \qquad & \vdots \\
\mathrm{T}\,\exists x\, A & & \mathrm{F}\,\forall x\, A \\
\vdots & & \vdots \\
| & & | \\
\mathrm{T}\, A[x/a] & & \mathrm{F}\, A[x/a]
\end{array}
$$

where a is a new parameter

Unsigned:

$$
\begin{array}{ccc}
\vdots & \qquad & \vdots \\
\forall x\, A & & \neg\,\exists x\, A \\
\vdots & & \vdots \\
| & & | \\
A[x/a] & & \neg\, A[x/a]
\end{array}
$$

where a is an arbitrary parameter

$$
\begin{array}{ccc}
\vdots & \qquad & \vdots \\
\exists x\, A & & \neg\,\forall x\, A \\
\vdots & & \vdots \\
| & & | \\
A[x/a] & & \neg\, A[x/a]
\end{array}
$$

where a is a new parameter

Remark 2.3.3. In the above tableau rules, "a is new" means that a does not occur in the path that is being extended. Or, we can insist that a not occur in the tableau that is being extended.

Remark 2.3.4. We are going to prove that the tableau method for predicate calculus is sound (Theorem 2.3.13) and complete (Theorem 2.5.5). In particular, a sentence A of the predicate calculus is logically valid if and only if there exists a finite closed signed tableau starting with $\mathrm{F}\, A$, or equivalently a finite closed unsigned tableau starting with $\neg\, A$.

Example 2.3.5. The signed tableau

$$\begin{array}{c}
\mathrm{F}\,(\exists x\,\forall y\,Rxy) \Rightarrow (\forall y\,\exists x\,Rxy) \\
\mathrm{T}\,\exists x\,\forall y\,Rxy \\
\mathrm{F}\,\forall y\,\exists x\,Rxy \\
\mathrm{T}\,\forall y\,Ray \\
\mathrm{F}\,\exists x\,Rxb \\
\mathrm{T}\,Rab \\
\mathrm{F}\,Rab
\end{array}$$

is closed. Therefore, by the Soundness Theorem, $(\exists x\,\forall y\,Rxy) \Rightarrow (\forall y\,\exists x\,Rxy)$ is logically valid.

Example 2.3.6. The unsigned tableau

$$\begin{array}{ccc}
\multicolumn{3}{c}{\neg\,((\exists x\,(Px \vee Qx)) \Leftrightarrow ((\exists x\,Px) \vee (\exists x\,Qx)))} \\
/ & & \backslash \\
\exists x\,(Px \vee Qx) & & \neg\,\exists x\,(Px \vee Qx) \\
\neg\,((\exists x\,Px) \vee (\exists x\,Qx)) & & (\exists x\,Px) \vee (\exists x\,Qx) \\
\neg\,\exists x\,Px & & \begin{array}{cc} / & \backslash \end{array} \\
\neg\,\exists x\,Qx & & \begin{array}{cc} \exists x\,Px & \exists x\,Qx \end{array} \\
Pa \vee Qa & & \begin{array}{cc} Pb & Qc \end{array} \\
\begin{array}{cc} / & \backslash \end{array} & & \begin{array}{cc} \neg\,(Pb \vee Qb) & \neg\,(Pc \vee Qc) \end{array} \\
\begin{array}{cc} Pa & Qa \end{array} & & \begin{array}{cc} \neg\,Pb & \neg\,Pc \end{array} \\
\begin{array}{cc} \neg\,Pa & \neg\,Qa \end{array} & & \begin{array}{cc} \neg\,Qb & \neg\,Qc \end{array}
\end{array}$$

is closed. Therefore, by the Soundness Theorem,

$$(\exists x\,(Px \vee Qx)) \Leftrightarrow ((\exists x\,Px) \vee (\exists x\,Qx))$$

is logically valid.

Exercises 2.3.7. Use signed tableaux to show that the following are logically valid.

1. $(\forall x\,(A \Rightarrow B)) \Rightarrow ((\forall x\,A) \Rightarrow (\forall x\,B))$

   Solution.

$$\begin{array}{c}
\mathrm{F}\,(\forall x\,(A \Rightarrow B)) \Rightarrow ((\forall x\,A) \Rightarrow (\forall x\,B)) \\
\mathrm{T}\ \forall x(A \Rightarrow B) \\
\mathrm{F}\,(\forall x\,A) \Rightarrow (\forall x\,B) \\
\mathrm{T}\ \forall x\,A \\
\mathrm{F}\ \forall x\,B \\
\mathrm{F}\,B[x/a] \\
\mathrm{T}\,A[x/a] \\
\mathrm{T}\,(A \Rightarrow B)[x/a] \\
\begin{array}{cc} / & \backslash \\ \mathrm{F}\,A[x/a] & \mathrm{T}\,B[x/a] \end{array}
\end{array}$$

2. $(\exists x\,(A \vee B)) \Leftrightarrow ((\exists x\,A) \vee (\exists x\,B))$

   Solution.

$$
\begin{array}{c}
\mathrm{F}\,(\exists x\,(A \vee B)) \Leftrightarrow ((\exists x\,A) \vee (\exists x\,B)) \\
\begin{array}{cc} / & \backslash \end{array} \\
\begin{array}{cc}
\begin{array}{c}
\mathrm{T}\ \exists x\,(A \vee B) \\
\mathrm{F}\,(\exists x\,A) \vee (\exists x\,B) \\
\mathrm{T}\,(A \vee B)[x/a] \\
\mathrm{F}\ \exists x\,A \\
\mathrm{F}\ \exists x\,B \\
\mathrm{F}\,A[x/a] \\
\mathrm{F}\,B[x/a] \\
\begin{array}{cc} / & \backslash \\ \mathrm{T}\,A[x/a] & \mathrm{T}\,B[x/a] \end{array}
\end{array}
&
\begin{array}{c}
\mathrm{F}\ \exists x\,(A \vee B) \\
\mathrm{T}\,(\exists x\,A) \vee (\exists x\,B) \\
\begin{array}{cc}
/ & \backslash \\
\begin{array}{c} \mathrm{T}\ \exists x\,A \\ \mathrm{T}\,A[x/a] \\ \mathrm{F}\,(A \vee B)[x/a] \\ \mathrm{F}\,A[x/a] \\ \mathrm{F}\,B[x/a] \end{array}
&
\begin{array}{c} \mathrm{T}\,\exists x\,B \\ \mathrm{T}\,B[x/a] \\ \mathrm{F}\,(A \vee B)[x/a] \\ \mathrm{F}\,A[x/a] \\ \mathrm{F}\,B[x/a] \end{array}
\end{array}
\end{array}
\end{array}
\end{array}
$$

3. $(\exists x\,A) \Leftrightarrow (\neg\,\forall x\,\neg\,A)$

   Solution.

$$
\begin{array}{c}
\mathrm{F}\,(\exists x\,A) \Leftrightarrow (\neg\,\forall x\,\neg\,A) \\
\begin{array}{cc}
/ & \backslash \\
\begin{array}{c}
\mathrm{T}\ \exists x\,A \\
\mathrm{F}\,\neg\,\forall x\,\neg\,A \\
\mathrm{T}\,A[x/a] \\
\mathrm{T}\ \forall x\,\neg\,A \\
\mathrm{T}\,(\neg\,A)[x/a] \\
\mathrm{F}\,A[x/a]
\end{array}
&
\begin{array}{c}
\mathrm{F}\ \exists x\,A \\
\mathrm{T}\,\neg\,\forall x\,\neg\,A \\
\mathrm{F}\ \forall x\,\neg\,A \\
\mathrm{F}\,(\neg\,A)[x/a] \\
\mathrm{T}\,A[x/a] \\
\mathrm{F}\,A[x/a]
\end{array}
\end{array}
\end{array}
$$

4. $(\forall x\,(A \vee C)) \Leftrightarrow ((\forall x\,A) \vee C)$, provided x is not free in C

   Solution.

$$
\begin{array}{c}
\mathrm{F}\,(\forall x\,(A \vee C)) \Leftrightarrow ((\forall x\,A) \vee C) \\
\begin{array}{cc}
/ & \backslash \\
\begin{array}{c}
\mathrm{T}\ \forall x\,(A \vee C) \\
\mathrm{F}\,(\forall x\,A) \vee C \\
\mathrm{F}\ \forall x\,A \\
\mathrm{F}\,C \\
\mathrm{F}\,A[x/a] \\
\mathrm{T}\,(A \vee C)[x/a] \\
\begin{array}{cc} / & \backslash \\ \mathrm{T}\,A[x/a] & \mathrm{T}\,C \end{array}
\end{array}
&
\begin{array}{c}
\mathrm{F}\ \forall x\,(A \vee C) \\
\mathrm{T}\,(\forall x\,A) \vee C \\
\mathrm{F}\,(A \vee C)[x/a] \\
\mathrm{F}\,A[x/a] \\
\mathrm{F}\,C \\
\begin{array}{cc} / & \backslash \\ \begin{array}{c} \mathrm{T}\ \forall x\,A \\ \mathrm{T}\,A[x/a] \end{array} & \mathrm{T}\,C \end{array}
\end{array}
\end{array}
\end{array}
$$

Exercise 2.3.8. Using the predicates Bx ("x is a barber in Podunk"), Cx ("x is a citizen of Podunk"), and Sxy ("x shaves y"), translate the following argument into a sentence of the predicate calculus.

> There is a barber in Podunk who shaves exactly those citizens of Podunk who do not shave themselves. Therefore, there is a barber in Podunk who is not a citizen of Podunk.

Use an unsigned tableau to test this argument for logical validity.

Solution. $(\exists x\,(Bx \wedge \forall y\,(Cy \Rightarrow (Sxy \Leftrightarrow \neg\, Syy)))) \Rightarrow \exists x\,(Bx \wedge \neg\, Cx)$. A tableau starting with the negation of this sentence (left to the reader) closes off to show that the sentence is logically valid.

Exercise 2.3.9. Translate the following argument into the predicate calculus, and use appropriate methods to establish its validity or invalidity.

> Anyone who can solve all logic problems is a good student. No student can solve every logic problem. Therefore, there are logic problems that no student can solve.

Exercise 2.3.10. Use an unsigned tableau to show that $\exists x\,(Px \Leftrightarrow \forall y\, Py)$ is logically valid.

The rest of this section is devoted to proving the Soundness Theorem 2.3.13.

Definition 2.3.11.

1. An *L-V-structure* consists of an L-structure M together with a mapping $\phi : V \to U_M$. If A is an L-V-sentence, we write

   $$A^\phi = A[a_1/\phi(a_1), \ldots, a_k/\phi(a_k)]$$

   where $a_1, \ldots, a_k$ are the parameters occurring in A. Note that A^ϕ is an L-U_M-sentence. Note also that, if A is an L-sentence, then $A^\phi = A$.

2. Let S be a finite or countable set of (signed or unsigned) L-V-sentences. An L-V-structure M, ϕ is said to *satisfy* S if $v_M(A^\phi) = \mathrm{T}$ for all $A \in S$. S is said to be *satisfiable*[6] if there exists an L-V-structure satisfying S. Note that this definition is compatible with Definition 2.2.7.

3. Let τ be an L-tableau. We say that τ is *satisfiable* if at least one path of τ is satisfiable.

Lemma 2.3.12. Let τ and τ' be tableaux such that τ' is an *immediate extension* of τ, i.e., τ' is obtained from τ by applying a tableau rule to a path of τ. If τ is satisfiable, then τ' is satisfiable.

Proof. The proof consists of one case for each tableau rule. We consider some representative cases.

[6]Similarly, the notions of logical validity and logical consequence are extended to L-V-sentences, in the obvious way, using L-V-structures. An L-V-sentence is said to be *logically valid* if it satisfied by all L-V-structures. An L-V-sentence is said to be a *logical consequence* of S if it is satisfied by all L-V-structures satisfying S.

Case 1. Suppose that τ' is obtained from τ by applying the rule

$$\begin{array}{c} \vdots \\ A \vee B \\ \vdots \\ / \quad \backslash \\ A \qquad B \end{array}$$

to the path θ in τ. Since τ is satisfiable, it has at least one satisfiable path, S. If $S \neq \theta$, then S is a path of τ', so τ' is satisfiable. If $S = \theta$, then θ is satisfiable, so let M and $\phi : V \to U_M$ satisfy θ. In particular $v_M((A \vee B)^\phi) = \mathrm{T}$, so we have at least one of $v_M(A^\phi) = \mathrm{T}$ and $v_M(B^\phi) = \mathrm{T}$. Thus M and ϕ satisfy at least one of θ, A and θ, B. Since these are paths of τ', it follows that τ' is satisfiable.

Case 2. Suppose that τ' is obtained from τ by applying the rule

$$\begin{array}{c} \vdots \\ \forall x\, A \\ \vdots \\ | \\ A[x/a] \end{array}$$

to the path θ in τ, where a is a parameter. Since τ is satisfiable, it has at least one satisfiable path, S. If $S \neq \theta$, then S is a path of τ', so τ' is satisfiable. If $S = \theta$, then θ is satisfiable, so let M and $\phi : V \to U_M$ satisfy θ. In particular $v_M(\forall x\,(A^\phi)) = v_M((\forall x\, A)^\phi) = \mathrm{T}$, so $v_M(A^\phi[x/c]) = \mathrm{T}$ for all $c \in U_M$. In particular

$$v_M(A[x/a]^\phi) = v_M(A^\phi[x/\phi(a)]) = \mathrm{T}.$$

Thus M and ϕ satisfy $\theta, A[x/a]$. Since this is a path of τ', it follows that τ' is satisfiable.

Case 3. Suppose that τ' is obtained from τ by applying the rule

$$\begin{array}{c} \vdots \\ \exists x\, A \\ \vdots \\ | \\ A[x/a] \end{array}$$

to the path θ in τ, where a is a new parameter. Since τ is satisfiable, it has at least one satisfiable path, S. If $S \neq \theta$, then S is a path of τ', so τ' is satisfiable. If $S = \theta$, then θ is satisfiable, so let M and $\phi : V \to U_M$ satisfy θ. In particular $v_M(\exists x\,(A^\phi)) = v_M((\exists x\, A)^\phi) = \mathrm{T}$, so $v_M(A^\phi[x/c]) = \mathrm{T}$ for at least one $c \in U_M$. Fix such a c and define $\phi' : V \to U_M$ by putting $\phi'(a) = c$, and $\phi'(b) = \phi(b)$ for all $b \neq a$, $b \in V$. Since a is new, we have $B^{\phi'} = B^\phi$ for all $B \in \theta$, and $A^{\phi'} = A^\phi$, hence $A[x/a]^{\phi'} = A^{\phi'}[x/\phi'(a)] = A^{\phi}[x/c]$. Thus $v_M(B^{\phi'}) = v_M(B^{\phi}) = \mathrm{T}$ for all $B \in \theta$, and $v_M(A[x/a]^{\phi'}) = v_M(A^{\phi}[x/c]) = \mathrm{T}$. Thus M and ϕ' satisfy $\theta, A[x/a]$. Since this is a path of τ', it follows that τ' is satisfiable. □

Theorem 2.3.13 (the Soundness Theorem)**.** Let $X_1, \ldots, X_k$ be a finite set of (signed or unsigned) sentences with parameters. If there exists a finite closed tableau starting with $X_1, \ldots, X_k$, then $X_1, \ldots, X_k$ is not satisfiable.

Proof. Let τ be a closed tableau starting with $X_1, \ldots, X_k$. Thus there is a finite sequence of tableaux $\tau_0, \tau_1, \ldots, \tau_n = \tau$ such that

$$\tau_0 = \begin{array}{c} X_1 \\ \vdots \\ X_k \end{array}$$

and each τ_{i+1} is an immediate extension of τ_i. Suppose $X_1, \ldots, X_k$ is satisfiable. Then τ_0 is satisfiable, and by induction on i using Lemma 2.3.12, it follows that all of the τ_i are satisfiable. In particular $\tau_n = \tau$ is satisfiable, but this is impossible since τ is closed. □

2.4 Logical Equivalence

Definition 2.4.1. Given an L-V-formula A, let $A' = A[x_1/a_1, \ldots, x_k/a_k]$, where $x_1, \ldots, x_k$ are the variables which occur freely in A, and $a_1, \ldots, a_k$ are parameters not occurring in A. Note that A' has no free variables, i.e., it is an L-V-sentence. We define A to be *satisfiable* if and only if A' is satisfiable, in the sense of Definition 2.3.11. We define A to be *logically valid* if and only if A' is logically valid, in the sense of Definition 2.3.11.

Exercises 2.4.2. Let A be an L-V-formula.

1. Show that A is logically valid if and only if $\neg A$ is not satisfiable. Show that A is satisfiable if and only if $\neg A$ is not logically valid.

2. Let x be a variable. Show that A is logically valid if and only if $\forall x\, A$ is logically valid. Show that A is satisfiable if and only if $\exists x\, A$ is satisfiable.

3. Let x be a variable, and let a be a parameter not occurring in A. Show that A is logically valid if and only if $A[x/a]$ is logically valid. Show that A is satisfiable if and only if $A[x/a]$ is satisfiable.

Definition 2.4.3. Let A and B be L-V-formulas. A and B are said to be *logically equivalent*, written $A \equiv B$, if $A \Leftrightarrow B$ is logically valid.

Exercise 2.4.4. Assume $A \equiv B$. Show that for any variable x, $\forall x\, A \equiv \forall x\, B$ and $\exists x\, A \equiv \exists x\, B$. Show that for any variable x and parameter a, $A[x/a] \equiv B[x/a]$.

Exercise 2.4.5. For an L-V-formula A, it is not in general true that $A \equiv A'$, where A' is as in Definition 2.4.1. Also, it is not in general true that $A \equiv \forall x\, A$, or that $A \equiv \exists x\, A$, or that $A \equiv A[x/a]$. Give examples illustrating these remarks.

Solution. Let A be the formula Px where P is a unary predicate. It is straightforward to show that $A \not\equiv A'$, $A \not\equiv \forall x\, A$, $A \not\equiv \exists x\, A$ and $A \not\equiv A[x/a]$.

Exercise 2.4.6. Given L-V-formulas A and B, let $A' = A[x_1/a_1, \ldots, x_k/a_k]$ and $B' = B[x_1/a_1, \ldots, x_k/a_k]$, where $x_1, \ldots, x_k$ are the variables occurring freely in A and B, and $a_1, \ldots, a_k$ are parameters not occurring in A or in B. Show that $A \equiv B$ if only if $A' \equiv B'$.

Remark 2.4.7. The results of Exercises 1.3.3 and 1.3.4 and Remark 1.3.5 for formulas of the propositional calculus, also hold for formulas of the predicate calculus. In particular, if $A_1 \equiv A_2$, then for any formula C containing A_1 as a part, if we replace one or more occurrences of the part A_1 by A_2, then the resulting formula is logically equivalent to C.

Remark 2.4.8. Some useful logical equivalences are:

1. (a) $\forall x\, A \equiv A$, provided x does not occur freely in A
   (b) $\exists x\, A \equiv A$, provided x does not occur freely in A
   (c) $\forall x\, A \equiv \forall y\, A[x/y]$, provided y does not occur in A
   (d) $\exists x\, A \equiv \exists y\, A[x/y]$, provided y does not occur in A

   Note that the last two equivalences provide for "replacement of bound variables". In this way, we can convert any formula into a logically equivalent formula with the following properties: no variable occurs both free and bound, and each bound variable is bound by at most one quantifier.

2. (a) $\forall x\, (A \wedge B) \equiv (\forall x\, A) \wedge (\forall x\, B)$
   (b) $\exists x\, (A \vee B) \equiv (\exists x\, A) \vee (\exists x\, B)$
   (c) $\exists x\, (A \Rightarrow B) \equiv (\forall x\, A) \Rightarrow (\exists x\, B)$

   Note however that, in general, $\exists x\, (A \wedge B) \not\equiv (\exists x\, A) \wedge (\exists x\, B)$, and $\forall x\, (A \vee B) \not\equiv (\forall x\, A) \vee (\forall x\, B)$, and $\forall x\, (A \Rightarrow B) \not\equiv (\exists x\, A) \Rightarrow (\forall x\, B)$.

   On the other hand, we have:

3. (a) $\exists x\, (A \wedge B) \equiv A \wedge (\exists x\, B)$, provided x does not occur freely in A
   (b) $\exists x\, (A \wedge B) \equiv (\exists x\, A) \wedge B$, provided x does not occur freely in B
   (c) $\forall x\, (A \vee B) \equiv A \vee (\forall x\, B)$, provided x does not occur freely in A
   (d) $\forall x\, (A \vee B) \equiv (\forall x\, A) \vee B$, provided x does not occur freely in B
   (e) $\exists x\, (A \Rightarrow B) \equiv A \Rightarrow (\exists x\, B)$, provided x does not occur freely in A
   (f) $\forall x\, (A \Rightarrow B) \equiv A \Rightarrow (\forall x\, B)$, provided x does not occur freely in A
   (g) $\exists x\, (A \Rightarrow B) \equiv (\forall x\, A) \Rightarrow B$, provided x does not occur freely in B

(h) $\forall x\,(A \Rightarrow B) \equiv (\exists x\, A) \Rightarrow B$, provided x does not occur freely in B

4. (a) $\exists x \neg A \equiv \neg \forall x\, A$

   (b) $\forall x \neg A \equiv \neg \exists x\, A$

   (c) $\forall x\, A \equiv \neg \exists x \neg A$

   (d) $\exists x\, A \equiv \neg \forall x \neg A$

Definition 2.4.9 (variants)**.** Let A be an L-FV-formula. A *variant* of A is any formula $\widehat{A}$ obtained from A by replacing some or all the bound variables of A by distinct new variables. Note that $\widehat{A}$ has the same free variables as A and is logically equivalent to A, in view of Remark 2.4.8, parts 1(c) and 1(d).

Example 2.4.10. Let A be $\forall x\, \exists y\, Rxyz$, and let $\widehat{A}$ be $\forall u\, \exists v\, Ruvz$. Then $\widehat{A}$ is a variant of A, hence $\widehat{A} \equiv A$.

Definition 2.4.11 (prenex form)**.** A formula is said to be *quantifier-free* if it contains no quantifiers. A formula is said to be *in prenex form* if it is of the form $Q_1 x_1 \cdots Q_n x_n\, B$, where each Q_i is a quantifier ($\forall$ or $\exists$), each x_i is a variable, and B is quantifier-free.

Example 2.4.12. The sentence

$$\forall x\, \forall y\, \exists z\, \forall w\, (Rxy \Rightarrow (Rxz \land Rzy \land \neg (Rzw \land Rwy)))$$

is in prenex form.

Exercise 2.4.13. Show that every L-V-formula is logically equivalent to an L-V-formula in prenex form. (Hint: Use the equivalences of Remark 2.4.8. First replace the given formula by a variant in which each bound variable is quantified only once and does not occur freely. Then use parts 3 and 4 to move all quantifiers to the front.)

Example 2.4.14. Consider the sentence $(\exists x\, Px) \land (\exists x\, Qx)$. We wish to put this into prenex form. Applying the equivalences of Remark 2.4.8, we have

$$\begin{aligned} (\exists x\, Px) \land (\exists x\, Qx) &\equiv (\exists x\, Px) \land (\exists y\, Qy) \\ &\equiv \exists x\, (Px \land (\exists y\, Qy)) \\ &\equiv \exists x\, \exists y\, (Px \land Qy) \end{aligned}$$

and this is in prenex form.

Exercise 2.4.15. Find a sentence in prenex normal form which is logically equivalent to $(\forall x\, \exists y\, Rxy) \Rightarrow \neg \exists x\, Px$.

Solution. $\exists x\, \forall y\, \forall z\, (Rxy \Rightarrow \neg Pz)$.

Exercises 2.4.16. Let A and B be L-V-formulas. Put the following into prenex form.

1. $(\exists x\, A) \wedge (\exists x\, B)$

2. $(\forall x\, A) \Leftrightarrow (\forall x\, B)$

3. $(\forall x\, A) \Leftrightarrow (\exists x\, B)$

Definition 2.4.17 (universal closure)**.** Let A be an L-V-formula. The *universal closure* of A is the L-V-sentence $A^* = \forall x_1 \cdots \forall x_k\, A$, where $x_1, \ldots, x_k$ are the variables which occur freely in A. Note that $A^{**} \equiv A^*$.

Exercises 2.4.18. Let A be an L-V-formula.

1. Show that A is logically valid if and only if A^*, the universal closure of A, is logically valid.

2. It is not true in general that $A \equiv A^*$. Give an example illustrating this.

3. It is not true in general that A is satisfiable if and only if A^* is satisfiable. Give an example illustrating this.

4. For L-V-formulas A and B, it is not true in general that $A \equiv B$ if and only if $A^* \equiv B^*$. Give an example illustrating this.

For completeness we state the following definition.

Definition 2.4.19. Let $A_1, \ldots, A_k, B$ be L-V-formulas. We say that B is a *logical consequence* of $A_1, \ldots, A_k$ if $(A_1 \wedge \cdots \wedge A_k) \Rightarrow B$ is logically valid. This is equivalent to saying that the universal closure of $(A_1 \wedge \cdots \wedge A_k) \Rightarrow B$ is logically valid.

Remark 2.4.20. A and B are logically equivalent if and only if each is a logical consequence of the other. A is logically valid if and only if A is a logical consequence of the empty set. $\exists x\, A$ is a logical consequence of $A[x/a]$, but the converse does not hold in general. $A[x/a]$ is a logical consequence of $\forall x\, A$, but the converse does not hold in general.

2.5 The Completeness Theorem

Let U be a nonempty set, and let S be a set of (signed or unsigned) L-U-sentences.

Definition 2.5.1. S is *closed* if S contains a conjugate pair of L-U-sentences. In other words, for some L-U-sentence A, S contains $\mathrm{T}\, A$, $\mathrm{F}\, A$ in the signed case, A, $\neg\, A$ in the unsigned case. S is *open* if it is not closed.

Definition 2.5.2. S is *U-replete* if S "obeys the tableau rules" with respect to U. We list some representative clauses of the definition.

1. If S contains $\mathrm{T} \neg A$, then S contains $\mathrm{F}\, A$. If S contains $\mathrm{F} \neg A$, then S contains $\mathrm{T}\, A$. If S contains $\neg \neg A$, then S contains A.

2. If S contains $\mathrm{T}\,A \wedge B$, then S contains both $\mathrm{T}\,A$ and $\mathrm{T}\,B$. If S contains $\mathrm{F}\,A \wedge B$, then S contains at least one of $\mathrm{F}\,A$ and $\mathrm{F}\,B$. If S contains $A \wedge B$, then S contains both A and B. If S contains $\neg\,(A \wedge B)$, then S contains at least one of $\neg\,A$ and $\neg\,B$.

3. If S contains $\mathrm{T}\,\exists x\, A$, then S contains $\mathrm{T}\,A[x/a]$ for at least one $a \in U$. If S contains $\mathrm{F}\,\exists x\, A$, then S contains $\mathrm{F}\,A[x/a]$ for all $a \in U$. If S contains $\exists x\, A$, then S contains $A[x/a]$ for at least one $a \in U$. If S contains $\neg\,\exists x\, A$, then S contains $\neg\,A[x/a]$ for all $a \in U$.

4. If S contains $\mathrm{T}\,\forall x\, A$, then S contains $\mathrm{T}\,A[x/a]$ for all $a \in U$. If S contains $\mathrm{F}\,\forall x\, A$, then S contains $\mathrm{F}\,A[x/a]$ for at least one $a \in U$. If S contains $\forall x\, A$, then S contains $A[x/a]$ for all $a \in U$. If S contains $\neg\,\forall x\, A$, then S contains $\neg\,A[x/a]$ for at least one $a \in U$.

Lemma 2.5.3 (Hintikka's Lemma)**.** If S is U-replete and open[7], then S is satisfiable. In fact, S is satisfiable in the domain U.

Proof. Assume S is U-replete and open. We define an L-structure M by putting $U_M = U$ and, for each n-ary predicate P of L,

$$P_M = \{\langle a_1, \ldots, a_n \rangle \in U^n \mid \mathrm{T}\,Pa_1 \cdots a_n \in S\}$$

in the signed case, and

$$P_M = \{\langle a_1, \ldots, a_n \rangle \in U^n \mid Pa_1 \cdots a_n \in S\}$$

in the unsigned case.

We claim that for all L-U-sentences A,

(a) if S contains $\mathrm{T}\,A$, then $v_M(A) = \mathrm{T}$

(b) if S contains $\mathrm{F}\,A$, then $v_M(A) = \mathrm{F}$

in the signed case, and

(c) if S contains A, then $v_M(A) = \mathrm{T}$

(d) if S contains $\neg\,A$, then $v_M(A) = \mathrm{F}$

in the unsigned case.

In both cases, the claim is easily proved by induction on the degree of A. We give the proof for some representative cases.

1. $\deg(A) = 0$. In this case A is atomic, say $A = Pa_1 \cdots a_n$.

   (a) If S contains $\mathrm{T}\,Pa_1 \cdots a_n$, then by definition of M we have $(a_1, \ldots, a_n) \in P_M$, so $v_M(Pa_1 \cdots a_n) = \mathrm{T}$.

[7]See also Exercise 2.5.7.

   (b) If S contains $\mathrm{F}\,Pa_1 \cdots a_n$, then S does not contain $\mathrm{T}\,Pa_1 \cdots a_n$ since S is open. Thus by definition of M we have $(a_1, \ldots, a_n) \notin P_M$, so $v_M(Pa_1 \cdots a_n) = \mathrm{F}$.

   (c) If S contains $Pa_1 \cdots a_n$, then by definition of M we have $(a_1, \ldots, a_n) \in P_M$, so $v_M(Pa_1 \cdots a_n) = \mathrm{T}$.

   (d) If S contains $\neg\, Pa_1 \cdots a_n$, then S does not contain $Pa_1 \cdots a_n$ since S is open. Thus by definition of M we have $(a_1, \ldots, a_n) \notin P_M$, so $v_M(Pa_1 \cdots a_n) = \mathrm{F}$.

2. $\deg(A) > 0$ and $A = \neg\, B$. Note that $\deg(B) < \deg(A)$ so the inductive hypothesis applies to B.

3. $\deg(A) > 0$ and $A = B \wedge C$. Note that $\deg(B)$ and $\deg(C)$ are $< \deg(A)$ so the inductive hypothesis applies to B and C.

   (a) If S contains $\mathrm{T}\,B \wedge C$, then by repleteness of S we see that S contains both $\mathrm{T}B$ and $\mathrm{T}C$. Hence by inductive hypothesis we have $v_M(B) = v_M(C) = \mathrm{T}$. Hence $v_M(B \wedge C) = \mathrm{T}$.

   (b) If S contains $\mathrm{F}\,B \wedge C$, then by repleteness of S we see that S contains at least one of $\mathrm{F}B$ and $\mathrm{F}C$. Hence by inductive hypothesis we have at least one of $v_M(B) = \mathrm{F}$ and $v_M(C) = \mathrm{F}$. Hence $v_M(B \wedge C) = \mathrm{F}$.

   (c) If S contains $B \wedge C$, then by repleteness of S we see that S contains both B and C. Hence by inductive hypothesis we have $v_M(B) = v_M(C) = \mathrm{T}$. Hence $v_M(B \wedge C) = \mathrm{T}$.

   (d) If S contains $\neg\,(B \wedge C)$, then by repleteness of S we see that S contains at least one of $\neg\, B$ and $\neg\, C$. Hence by inductive hypothesis we have at least one of $v_M(B) = \mathrm{F}$ and $v_M(C) = \mathrm{F}$. Hence $v_M(B \wedge C) = \mathrm{F}$.

4. $\deg(A) > 0$ and $A = \exists x\, B$. Note that for all $a \in U$ we have $\deg(B[x/a]) < \deg(A)$, so the inductive hypothesis applies to $B[x/a]$.

5. $\deg(A) > 0$ and $A = \forall x\, B$. Note that for all $a \in U$ we have $\deg(B[x/a]) < \deg(A)$, so the inductive hypothesis applies to $B[x/a]$.

$\square$

We shall now use Hintikka's Lemma to prove the completeness of the tableau method. As in Section 2.3, Let $V = \{a_1, \ldots, a_n, \ldots\}$ be the set of parameters. Recall that a tableau is a tree whose nodes carry L-V-sentences.

Lemma 2.5.4. Let τ_0 be a finite tableau. By applying tableau rules, we can extend τ_0 to a (possibly infinite) tableau τ with the following properties: every closed path of τ is finite, and every open path of τ is V-replete.

Proof. The idea is to start with τ_0 and use tableau rules to construct a sequence of finite extensions $\tau_0, \tau_1, \ldots, \tau_i, \ldots$. If some τ_i is closed, then the construction halts, i.e., $\tau_j = \tau_i$ for all $j \geq i$, and we set $\tau = \tau_i$. In any case, we set $\tau = \tau_\infty = \bigcup_{i=0}^{\infty} \tau_i$. In the course of the construction, we apply tableau rules systematically to ensure that τ_∞ will have the desired properties, using the fact that $V = \{a_1, a_2, \ldots, a_n, \ldots\}$ is countably infinite.

Here are the details of the construction. Call a node X of τ_i *quasiuniversal* if it is of the form $\mathrm{T}\,\forall x\, A$ or $\mathrm{F}\,\exists x\, A$ or $\forall x\, A$ or $\neg\,\exists x\, A$. Our construction begins with τ_0. Suppose we have constructed τ_{2i}. For each quasiuniversal node X of τ_{2i} and each $n \leq 2i$, apply the appropriate tableau rule to extend each open path of τ_{2i} containing X by $\mathrm{T}\,A[x/a_n]$ or $\mathrm{F}\,A[x/a_n]$ or $A[x/a_n]$ or $\neg\,A[x/a_n]$ as the case may be. Let τ_{2i+1} be the finite tableau so obtained. Next, for each non-quasiuniversal node X of τ_{2i+1}, extend each open path containing X by applying the appropriate tableau rule. Again, let τ_{2i+2} be the finite tableau so obtained.

In this construction, a closed path is never extended, so all closed paths of τ_∞ are finite. In addition, the construction ensures that each open path of τ_∞ is V-replete. Thus τ_∞ has the desired properties. This proves our lemma. $\square$

Theorem 2.5.5 (the Completeness Theorem)**.** Let $X_1, \ldots, X_k$ be a finite set of (signed or unsigned) sentences with parameters. If $X_1, \ldots, X_k$ is not satisfiable, then there exists a finite closed tableau starting with $X_1, \ldots, X_k$. If $X_1, \ldots, X_k$ is satisfiable, then $X_1, \ldots, X_k$ is satisfiable in the domain V.

Proof. By Lemma 2.5.4 there exists a (possibly infinite) tableau τ starting with $X_1, \ldots, X_k$ such that every closed path of τ is finite, and every open path of τ is V-replete. If τ is closed, then by König's Lemma (Theorem 1.6.6), τ is finite. If τ is open, let S be an open path of τ. Then S is V-replete. By Hintikka's Lemma 2.5.3, S is satisfiable in V. Hence $X_1, \ldots, X_k$ is satisfiable in V. $\square$

Definition 2.5.6. Let L, U, and S be as in Definition 2.5.1. S is said to be *atomically closed* if S contains a conjugate pair of atomic L-U-sentences. In other words, for some n-ary L-predicate P and $a_1, \ldots, a_n \in U$, S contains $\mathrm{T}\,Pa_1 \cdots a_n$, $\mathrm{F}\,Pa_1 \cdots a_n$ in the signed case, and $Pa_1 \cdots a_n$, $\neg\,Pa_1 \cdots a_n$ in the unsigned case. S is *atomically open* if it is not atomically closed.

Exercise 2.5.7. Show that Lemmas 2.5.3 and 2.5.4 and Theorem 2.5.5 continue to hold with "closed" ("open") replaced by "atomically closed" ("atomically open").

Remark 2.5.8. Corollaries 1.5.9, 1.5.10, 1.5.11 carry over from the propositional calculus to the predicate calculus. In particular, the tableau method provides a test for logical validity of sentences of the predicate calculus.

Note however that the test is only partially effective. If a sentence A is logically valid, we will certainly find a finite closed tableau starting with $\neg\,A$. But if A is not logically valid, we will not necessarily find a finite tableau which demonstrates this. See the following example.

Example 2.5.9. In 2.2.12 we have seen an example of a sentence A_∞ which is satisfiable in a countably infinite domain but not in any finite domain. It is instructive to generate a tableau starting with A_∞.

$$
\begin{array}{c}
A_\infty \\
\vdots \\
\forall x\,\forall y\,\forall z\,((Rxy \wedge Ryz) \Rightarrow Rxz) \\
\forall x\,\forall y\,(Rxy \Rightarrow \neg\, Ryx) \\
\forall x\,\exists y\,Rxy \\
\exists y\,Ra_1y \\
Ra_1a_2 \\
\forall y\,(Ra_1y \Rightarrow \neg\, Rya_1) \\
Ra_1a_2 \Rightarrow \neg\, Ra_2a_1 \\
\begin{array}{cc}
/ & \backslash \\
\neg\, Ra_1a_2 & \begin{array}{c}
\neg\, Ra_2a_1 \\
\exists y\,Ra_2y \\
Ra_2a_3 \\
\vdots \\
\neg\, Ra_3a_2 \\
\forall y\,\forall z\,((Ra_1y \wedge Ryz) \Rightarrow Ra_1z) \\
\forall z\,((Ra_1a_2 \wedge Ra_2z) \Rightarrow Ra_1z) \\
(Ra_1a_2 \wedge Ra_2a_3) \Rightarrow Ra_1a_3 \\
\begin{array}{cc}
/ & \backslash \\
\neg\,(Ra_1a_2 \wedge Ra_2a_3) & Ra_1a_3 \\
\begin{array}{cc}
/ & \backslash \\
\neg\, Ra_1a_2 & \neg\, Ra_2a_3
\end{array} & \begin{array}{c}
\vdots \\
\neg\, Ra_3a_1 \\
\exists y\,Ra_3y \\
Ra_3a_4 \\
\vdots
\end{array}
\end{array}
\end{array}
\end{array}
\end{array}
$$

An infinite open path gives rise (via the proof of Hintikka's Lemma) to an infinite L-structure M with $U_M = \{a_1, a_2, \ldots, a_n, \ldots\}$, $R_M = \{\langle a_m, a_n\rangle \mid 1 \leq m < n\}$. Clearly $M \models A_\infty$.

Remark 2.5.10. In the course of applying a tableau test, we will sometimes find a finite open path which is U-replete for some finite set of parameters $U \subseteq V$. In this case, the proof of Hintikka's Lemma provides a finite L-structure with domain U.

Example 2.5.11. Let A be the sentence $(\forall x\,(Px \vee Qx)) \Rightarrow ((\forall x\,Px) \vee (\forall x\,Qx))$.

Testing A for logical validity, we have:

$$
\begin{array}{c}
\neg\, A \\
\forall x\,(Px \lor Qx) \\
\neg\,((\forall x\, Px) \lor (\forall x\, Qx)) \\
\neg\,\forall x\, Px \\
\neg\,\forall x\, Qx \\
\neg\, Pa \\
\neg\, Qb \\
Pa \lor Qa \\
Pb \lor Qb \\
\begin{array}{ccc}
/ & & \backslash \\
Pa & & Qa \\
 & & \begin{array}{cc} / & \backslash \\ Pb & Qb \end{array}
\end{array}
\end{array}
$$

This tableau has a unique open path, which gives rise (via the proof of Hintikka's Lemma) to a finite L-structure M with $U_M = \{a, b\}$, $P_M = \{b\}$, $Q_M = \{a\}$. Clearly M falsifies A.

Exercise 2.5.12. Using the predicate Rxy ("x is an ancestor of y"), translate the following argument into a sentence of the predicate calculus.

> Every ancestor of an ancestor of an individual is an ancestor of the same individual. No individual is his own ancestor. Therefore, there is an individual who has no ancestor.

Is this argument valid? Justify your answer by means of an appropriate structure or tableau.

Solution. $((\forall x\,\forall y\,((\exists z\,(Rxz \land Rzy)) \Rightarrow Rxy)) \land \neg\,\exists x\, Rxx) \Rightarrow \exists x\,\neg\,\exists y\, Ryx$.
A tableau starting with the negation of this sentence (left to the reader) fails to close off. The structure $(\mathbb{N}, >_{\mathbb{N}})$ falsifies the sentence, thus showing that it is not logically valid.

Exercise 2.5.13. Using the predicates Sx ("x is a set") and Exy ("x is a member of y"), translate the following into a sentence of the predicate calculus.

> There exists a set whose members are exactly those sets which are not members of themselves.

Use an unsigned tableau to test your sentence for *consistency*, i.e., satisfiability.

Exercise 2.5.14. Using the predicates Sx ("x is Socrates"), Hx ("x is a man"), Mx ("x is mortal"), translate the following argument into a sentence of the predicate calculus.

> Socrates is a man. All men are mortal. Therefore, Socrates is mortal.

Use an unsigned tableau to test whether the argument is valid.

Exercises 2.5.15.

1. Using the predicates Sx ("x can solve this problem"), Mx ("x is a mathematician"), Jx ("x is Joe"), translate the following argument into a sentence of the predicate calculus.

   > If anyone can solve this problem, some mathematician can solve it. Joe is a mathematician and cannot solve it. Therefore, nobody can solve it.

   Use an unsigned tableau to test whether the argument is valid.

2. Using the same predicates as above, translate the following argument into a sentence of the predicate calculus.

   > Any mathematician can solve this problem if anyone can. Joe is a mathematician and cannot solve it. Therefore, nobody can solve it.

   Use an unsigned tableau to test whether the argument is valid.

2.6 The Compactness Theorem

Theorem 2.6.1 (the Compactness Theorem, countable case)**.** Let S be a countably infinite set of sentences of the predicate calculus. S is satisfiable if and only if each finite subset of S is satisfiable.

Proof. We combine the ideas of the proofs of the Countable Compactness Theorem for propositional calculus (Theorem 1.7.1) and the Completeness Theorem for predicate calculus (Theorem 2.5.5).

Details: Let $S = \{A_0, A_1, \ldots, A_i, \ldots\}$. Start by letting τ_0 be the empty tableau. Suppose we have constructed τ_{2i}. Extend τ_{2i} to τ'_{2i} by appending A_i to each open path of τ_{2i}. Since $\{A_0, A_1, \ldots, A_i\}$ is satisfiable, τ'_{2i} has at least one open path. Now extend τ'_{2i} to τ_{2i+1} and then to τ_{2i+2} as in the proof of Lemma 2.5.4. Finally put $\tau = \tau_\infty = \bigcup_{i=1}^{\infty} \tau_i$. As in Lemma 2.5.4 we have that every closed path of τ is finite, and every open path of τ is V-replete. Note also that τ is an infinite, finitely branching tree. By König's Lemma (Theorem 1.6.6), let S' be an infinite path in τ. Then S' is a V-replete open set which contains S. By Hintikka's Lemma for the predicate calculus (Lemma 2.5.3), S' is satisfiable. Hence S is satisfiable. $\square$

Theorem 2.6.2 (the Compactness Theorem, uncountable case)**.** Let S be an uncountable set of sentences of the predicate calculus. S is satisfiable if and only if each finite subset of S is satisfiable.

Proof. Assume that S is finitely satisfiable. For each sentence $A \in S$ of the form $\exists x\, B$ or $\neg\forall x\, B$, introduce a new parameter c_A. Let U_S be the set of parameters so introduced. Let S' be S together with the sentences $B[x/c_A]$ or $\neg B[x/c_A]$

as the case may be, for all $c_A \in U_S$. Then S' is a set of L-U_S-sentences, and it is easy to verify that S' is finitely satisfiable. By Zorn's Lemma, let S'' be a maximal finitely satisfiable set of L-U_S-sentences extending S'.

Now inductively define $S_0 = S$, $S_{n+1} = S_n''$, $S_\infty = \bigcup_{n=0}^\infty S_n$, $U = \bigcup_{n=0}^\infty U_{S_n}$. It is straightforward to verify that S_∞ is U-replete and open. Hence, by Hintikka's Lemma, S_∞ is satisfiable in the domain U. Since $S \subseteq S_\infty$, it follows that S is satisfiable in U. □

Exercise 2.6.3. Let L be a language consisting of a binary predicate R and some additional predicates. Let $M = (U_M, R_M, \ldots)$ be an L-structure such that (U_M, R_M) is isomorphic to $(\mathbb{N}, <_\mathbb{N})$. Note that M contains no infinite R-descending sequence. Show that there exists an L-structure M' such that:

1. M and M' satisfy the same L-sentences.
2. M' contains an infinite R-descending sequence. In other words, there exist elements $a_1', a_2', \ldots, a_n', \ldots \in U_{M'}$ such that $\langle a_{n+1}', a_n' \rangle \in R_{M'}$ for all $n = 1, 2, \ldots$.

Hint: Use the Compactness Theorem.

Exercise 2.6.4. Generalize Exercise 2.6.3 replacing $(\mathbb{N}, <_\mathbb{N})$ by an arbitrary infinite linear ordering with no infinite descending sequence. Show that M' can be obtained such that $(U_{M'}, R_{M'})$ is a linear ordering which contains an infinite descending sequence.

Exercise 2.6.5. Let $L = \{R, \ldots\}$ be a language which includes a binary predicate R. Let S be a set of L-sentences. Assume that for each $n \geq 1$ there exists an L-structure $(U_n, R_n, \ldots)$ satisfying S and containing elements $a_{n1}, \ldots, a_{nn}$ such that $\langle a_{ni}, a_{nj} \rangle \in R_n$ for all i and j with $1 \leq i < j \leq n$. Prove that there there exists an L-structure $(U_\infty, R_\infty, \ldots)$ satisfying S and containing elements $a_{\infty i}$, $i = 1, 2, \ldots$ such that $\langle a_{\infty i}, a_{\infty j} \rangle \in R_\infty$ for all i and j with $1 \leq i < j$.

Solution. Let $L^* = L \cup \{P_1, P_2, \ldots\}$ where $P_1, P_2, \ldots$ are new unary predicates. Let S^* be S plus $\exists x\, P_i x$ plus $\forall x\, \forall y\, ((P_i x \wedge P_j y) \Rightarrow Rxy)$, $1 \leq i < j$. Consider the L^*-structures $(U_n, R_n, \ldots, P_{n1}, \ldots, P_{nn}, \ldots)$, $n \geq 1$, where $P_{ni} = \{a_{ni}\}$ for $1 \leq i \leq n$, and $P_{ni} = \{\}$ for $i > n$. Clearly each finite subset of S^* is satisfied by all but finitely many of these structures. It follows by the Compactness Theorem that S^* is satisfiable. Let $(U_\infty, R_\infty, \ldots, P_{\infty 1}, P_{\infty 2}, \ldots)$ be an L^*-structure satisfying S^*. Clearly the L-structure $(U_\infty, R_\infty, \ldots)$ has the desired properties.

2.7 Satisfiability in a Domain

The notion of satisfiability in a domain was introduced in Definition 2.2.9.

Theorem 2.7.1. Let S be a set of L-sentences.

1. Assume that S is finite or countably infinite. If S is satisfiable, then S is satisfiable in a countably infinite domain.

2. Assume that S is of cardinality $\kappa \geq \aleph_0$. If S is satisfiable, then S is satisfiable in a domain of cardinality κ.

Proof. Parts 1 and 2 follow easily from the proofs of Compactness Theorems 2.6.1 and 2.6.2, respectively. In the countable case we have that S is satisfiable in V, which is countably infinite. In the uncountable case we have that S is satisfiable in U, where U is as in the proof of 2.6.2. By the arithmetic of infinite cardinal numbers, the cardinality of U is $\kappa \cdot \aleph_0 = \kappa$. □

In Example 2.2.12 we have seen a sentence A_∞ which is satisfiable in a countably infinite domain but not in any finite domain. Regarding satisfiability in finite domains, we have:

Example 2.7.2. Given a positive integer n, we exhibit a sentence A_n which is satisfiable in a domain of cardinality n but not in any domain of smaller cardinality. Our sentence A_n is $(1) \wedge (2) \wedge (3)$ with

$$\begin{array}{ll} (1) & \forall x\, \forall y\, \forall z\, ((Rxy \wedge Ryz) \Rightarrow Ryz) \\ (2) & \forall x\, \forall y\, (Rxy \Rightarrow \neg\, Ryx) \\ (3) & \exists x_1 \cdots \exists x_n\, (Rx_1x_2 \wedge Rx_2x_3 \wedge \cdots \wedge Rx_{n-1}x_n) \end{array}$$

On the other hand, we have:

Theorem 2.7.3. Let M and M' be L-structures. Assume that there exists an onto mapping $\phi : U_M \to U_{M'}$ such that for all n-ary predicates P of L and all n-tuples $\langle a_1, \ldots, a_n \rangle \in (U_M)^n$, $\langle a_1, \ldots, a_n \rangle \in P_M$ if and only if $\langle \phi(a_1), \ldots, \phi(a_n) \rangle \in P_{M'}$. Then as in Theorem 2.2.6 we have $v_M(A) = v_{M'}(A')$ for all L-U_M-sentences A, where $A' = A[a_1/\phi(a_1), \ldots, a_k/\phi(a_k)]$. In particular, M and M' satisfy the same L-sentences.

Proof. The proof is by induction on the degree of A. Suppose for example that $A = \forall x\, B$. Then by definition of v_M we have that $v_M(A) = \mathrm{T}$ if and only if $v_M(B[x/a]) = \mathrm{T}$ for all $a \in U_M$. By inductive hypothesis, this holds if and only if $v_{M'}(B[x/a]') = \mathrm{T}$ for all $a \in U_M$. But for all $a \in U_M$ we have $B[x/a]' = B'[x/\phi(a)]$. Thus our condition is equivalent to $v_{M'}(B'[x/\phi(a)]) = \mathrm{T}$ for all $a \in U_M$. Since $\phi : U_M \to U_{M'}$ is onto, this is equivalent to $v_{M'}(B'[x/b]) = \mathrm{T}$ for all $b \in U_{M'}$. By definition of $v_{M'}$ this is equivalent to $v_{M'}(\forall x\, B') = \mathrm{T}$. But $\forall x\, B' = A'$, so our condition is equivalent to $v_{M'}(A') = \mathrm{T}$. □

Corollary 2.7.4. Let S be a set of L-sentences. If S is satisfiable in a domain U, then S is satisfiable in any domain of the same or larger cardinality.

Proof. Suppose S is satisfiable in domain U. Let U' be a set of cardinality greater than or equal to that of U. Let $\phi : U' \to U$ be onto. If M is any L-structure with $U_M = U$, we can define an L-structure M' with $U_{M'} = U'$ by putting $P_{M'} = \{\langle a_1, \ldots, a_n \rangle \mid \langle \phi(a_1), \ldots, \phi(a_n) \rangle \in P_M\}$ for all n-ary predicates P of L. By Theorem 2.7.3, M and M' satisfy the same L-sentences. In particular, if $M \models S$, then $M' \models S$. □

Remark 2.7.5. We shall see later[8] that Theorem 2.7.3 and Corollary 2.7.4 fail for normal satisfiability.

[8] See Section 4.1.

Chapter 3

Proof Systems for Predicate Calculus

3.1 Introduction to Proof Systems

Definition 3.1.1. An *abstract proof system* consists of a set $\mathfrak{X}$ together with a relation $\mathfrak{R} \subseteq \bigcup_{k=0}^{\infty} \mathfrak{X}^{k+1}$. Elements of $\mathfrak{X}$ are called *objects*. Elements of $\mathfrak{R}$ are called *rules of inference*. An object $X \in \mathfrak{X}$ is said to be *derivable*, or *provable*, if there exists a finite sequence of objects $X_1, \ldots, X_n$ such that $X_n = X$ and, for each $i \leq n$, there exist $j_1, \ldots, j_k < i$ such that $\langle X_{j_1}, \ldots, X_{j_k}, X_i \rangle \in \mathfrak{R}$. The sequence $X_1, \ldots, X_n$ is called a *derivation* of X, or a *proof* of X.

Notation 3.1.2. For $k \geq 1$ it is customary to write

$$\frac{X_1 \ \cdots \ X_k}{Y}$$

indicating that $\langle X_1, \ldots, X_k, Y \rangle \in \mathfrak{R}$. This is to be understood as "from the premises $X_1, \ldots, X_k$ we may immediately infer the conclusion Y". For $k = 0$ we may write

$$\frac{}{Y}$$

or simply Y, indicating that $\langle Y \rangle \in \mathfrak{R}$. This is to be understood as "we may immediately infer Y from no premises", or "we may assume Y".

Definition 3.1.3. Let L be a language. Recall that V is the set of parameters. A *Hilbert-style proof system* for L is a proof system with the following properties:

1. The objects are sentences with parameters. In other words,

$$\mathfrak{X} = \{A \mid A \text{ is an } L\text{-}V\text{-sentence}\}.$$

2. For each rule of inference

$$\frac{A_1 \ \cdots \ A_k}{B}$$

(i.e., $\langle A_1, \ldots, A_k, B \rangle \in \mathfrak{R}$), we have that B is a logical consequence of $A_1, \ldots, A_k$. This property is known as *soundness*. It implies that every L-V-sentence which is derivable is logically valid.

3. For all L-V-sentences A, B, we have a rule of inference $\langle A, A \Rightarrow B, B \rangle \in \mathfrak{R}$, i.e.,

$$\frac{A \quad A \Rightarrow B}{B}\,.$$

   In other words, from A and $A \Rightarrow B$ we immediately infer B. This collection of inference rules is known as *modus ponens*.

4. An L-V-sentence A is logically valid if and only if A is derivable. This property is known as *completeness*.

Remark 3.1.4. In Section 3.3 we shall exhibit a particular Hilbert-style proof system, LH. The soundness of LH will be obvious. In order to verify the completeness of LH, we shall first prove a result known as the Companion Theorem, which is also of interest in its own right.

3.2 The Companion Theorem

In this section we shall comment on the notion of logical validity for sentences of the predicate calculus. We shall analyze logical validity into two components: a propositional component (quasitautologies), and a quantificational component (companions).

Definition 3.2.1 (quasitautologies)**.**

1. A *tautology* is a propositional formula which is logically valid.

2. A *quasitautology* is an L-V-sentence of the form $F[p_1/A_1, \ldots, p_k/A_k]$, where F is a tautology, $p_1, \ldots, p_k$ are the atoms occurring in F, and $A_1, \ldots, A_k$ are L-V-sentences.

For example, $p \Rightarrow (q \Rightarrow p)$ is a tautology. This implies that, for all L-V-sentences A and B, $A \Rightarrow (B \Rightarrow A)$ is a quasitautology.

Remarks 3.2.2.

1. Obviously, every quasitautology is logically valid.

2. There is a decision procedure[1] for quasitautologies. One such decision procedure is based on truth tables. Another is based on propositional tableaux.

[1]In other words, there is a Turing algorithm which, given an L-V-sentence A as input, will eventually halt with output 1 if A is a quasitautology, 0 if A is not a quasitautology.

3. It can be shown that there is no decision procedure for logical validity. (This result is known as Church's Theorem.) Therefore, in relation to the problem of characterizing logical validity, we regard the quasitautologies as trivial.

Let A be an L-V-sentence.

Definition 3.2.3 (companions)**.** A *companion* of A is any L-V-sentence of one of the forms

$$
\begin{array}{ll}
(1) & (\forall x\, B) \Rightarrow B[x/a] \\
(2) & B[x/a] \Rightarrow (\forall x\, B) \\
(3) & (\exists x\, B) \Rightarrow B[x/a] \\
(4) & B[x/a] \Rightarrow (\exists x\, B)
\end{array}
$$

where, in (2) and (3), the parameter a may not occur in A or in B.

Lemma 3.2.4. Let C be a companion of A.

1. A is satisfiable if and only if $C \wedge A$ is satisfiable.

2. A is logically valid if and only if $C \Rightarrow A$ is logically valid.

Proof. Let C be a companion of A.

For part 1, assume that A is satisfiable. In accordance with Definition 2.3.11, let M, ϕ be an L-V-structure satisfying A. If C is of the form 3.2.3(1) or 3.2.3(4), then C is logically valid, hence M, ϕ satisfies $C \wedge A$. Next, consider the case when C is of the form 3.2.3(2). If M, ϕ satisfies $\forall x\, B$, then M, ϕ satisfies C. Otherwise we have $v_M(\forall x\, B^\phi) = \mathrm{F}$, so let $c \in U_M$ be such that $v_M(B^\phi[x/c]) = \mathrm{F}$. Define $\phi' : V \to U_M$ by putting $\phi'(a) = c$, $\phi'(b) = \phi(b)$ for $b \neq a$. Since a does not occur in A, we have that M, ϕ' satisfies A. Also, since a does not occur in B, we have $B[x/a]^{\phi'} = B^{\phi'}[x/c] = B^\phi[x/c]$, hence $v_M(B[x/a]^{\phi'}) = v_M(B^\phi[x/c]) = \mathrm{F}$, i.e., M, ϕ' satisfies $\neg B[x/a]$. Thus M, ϕ' satisfies $C \wedge A$. The case when C is of the form 3.2.3(3) is handled similarly.

For part 2 note that, since C is a companion of A, C is a companion of $\neg A$. Thus we have that A is logically valid if and only if $\neg A$ is not satisfiable, if and only if $C \wedge \neg A$ is not satisfiable (by part 1), if and only if $\neg(C \wedge \neg A)$ is logically valid, i.e., $C \Rightarrow A$ is logically valid. $\square$

Definition 3.2.5 (companion sequences)**.** A *companion sequence* of A is a finite sequence $C_1, \ldots, C_n$ such that, for each $i < n$, C_{i+1} is a companion of

$$(C_1 \wedge \cdots \wedge C_i) \Rightarrow A.$$

Lemma 3.2.6. If $C_1, \ldots, C_n$ is a companion sequence of A, then A is logically valid if and only if $(C_1 \wedge \cdots \wedge C_n) \Rightarrow A$ is logically valid.

Proof. Note that $(C_1 \wedge \cdots \wedge C_n) \Rightarrow A$ is quasitautologically equivalent to

$$C_n \Rightarrow (C_{n-1} \Rightarrow \cdots \Rightarrow (C_1 \Rightarrow A)).$$

Our lemma follows by n applications of part 2 of Lemma 3.2.4. $\square$

Theorem 3.2.7 (the Companion Theorem)**.** *A is logically valid if and only if there exists a companion sequence $C_1, \ldots, C_n$ of A such that*

$$(C_1 \land \cdots \land C_n) \Rightarrow A$$

is a quasitautology.

Proof. The "if" part is immediate from Lemma 3.2.6. For the "only if" part, assume that A is logically valid. By Theorem 2.5.5 let τ be a finite closed unsigned tableau starting with $\neg A$. Thus we have a finite sequence of tableaux $\tau_0, \tau_1, \ldots, \tau_n$ where $\tau_0 = \neg A$, $\tau_n = \tau$, and each τ_{i+1} is obtained by applying a tableau rule R_i to τ_i. If R_i is a quantifier rule, let C_i be an appropriate companion. Thus $C_1, \ldots, C_n$ is a companion sequence for A, and we can easily transform τ into a closed tableau τ' starting with

$$\begin{array}{c} \neg A \\ C_1 \\ \vdots \\ C_n \end{array}$$

in which only propositional tableau rules are applied. Thus $(C_1 \land \cdots \land C_n) \Rightarrow A$ is a quasitautology. This proves our theorem.

For instance, if R_i is the tableau rule

$$(*) \qquad \begin{array}{c} \vdots \\ \forall x\, B \\ \vdots \\ | \\ B[x/a] \end{array} ,$$

where a is an arbitrary parameter, let C_i be the companion $(\forall x\, B) \Rightarrow B[x/a]$, and replace the application of $(*)$ by

$$\begin{array}{ccc} & \vdots & \\ & (\forall x\, B) \Rightarrow B[x/a] & \\ & \vdots & \\ & \forall x\, B & \\ & \vdots & \\ / & & \backslash \\ \neg \forall x\, B & & B[x/a] \end{array}$$

noting that the left-hand path is closed.

Similarly, if R_i is the tableau rule

$$
(**) \qquad \begin{array}{c} \vdots \\ \neg\,\forall x\, B \\ \vdots \\ | \\ \neg\, B[x/a] \end{array}
$$

where a is a new parameter, let C_i be the companion $B[x/a] \Rightarrow (\forall x\, B)$, and replace the application of $(**)$ by

$$
\begin{array}{ccc} & \vdots & \\ & B[x/a] \Rightarrow (\forall x\, B) & \\ & \vdots & \\ & \neg\,\forall x\, B & \\ & \vdots & \\ & / \quad \backslash & \\ \neg\, B[x/a] & & \forall x\, B \end{array}
$$

noting that the right-hand path is closed. $\square$

Example 3.2.8. As an example illustrating Theorem 3.2.7 and its proof, let A be the sentence $(\exists x\,(Px \vee Qx)) \Rightarrow ((\exists x\, Px) \vee (\exists x\, Qx))$. Let τ be the closed tableau

$$
\begin{array}{ccc} & \neg\, A & \\ & \exists x(Px \vee Qx) & \\ & \neg\,((\exists x\, Px) \vee (\exists x\, Qx)) & \\ & \neg\,\exists x\, Px & \\ & \neg\,\exists x\, Qx & \\ & Pa \vee Qa & \\ & \neg\, Pa & \\ & \neg\, Qa & \\ & / \quad \backslash & \\ Pa & & Qa \end{array}
$$

which shows that A is logically valid. Examining the applications of quantifier rules in τ, we obtain the companion sequence C_1, C_2, C_3 for A, where C_1 is $(\exists x\,(Px \vee Qx)) \Rightarrow (Pa \vee Qa)$, C_2 is $Pa \Rightarrow \exists x\, Px$, C_3 is $Qa \Rightarrow \exists x\, Qx$. Clearly $(C_1 \wedge C_2 \wedge C_3) \Rightarrow A$ is a quasitautology.

Exercise 3.2.9. Let A be the logically valid sentence $\exists x\,(Px \Rightarrow \forall y\, Py)$. Find a companion sequence $C_1, \ldots, C_n$ for A such that $(C_1 \wedge \cdots \wedge C_n) \Rightarrow A$ is a quasitautology.

Solution. The unsigned tableau

$$\begin{array}{c} \neg\,\exists x\,(Px \Rightarrow \forall y\,Py) \\ \neg\,(Pa \Rightarrow \forall y\,Py) \\ Pa \\ \neg\,\forall y\,Py \\ \neg\,Pb \\ \neg\,(Pb \Rightarrow \forall y\,Py) \\ Pb \end{array}$$

is closed and shows that A is logically valid. From this tableau we read off the companion sequence C_1, C_2, C_3, where

$$\begin{array}{lll} C_1 & \text{is} & (Pa \Rightarrow \forall y\,Py) \Rightarrow \exists x\,(Px \Rightarrow \forall y\,Py), \\ C_2 & \text{is} & Pb \Rightarrow \forall y\,Py, \\ C_3 & \text{is} & (Pb \Rightarrow \forall y\,Py) \Rightarrow \exists x\,(Px \Rightarrow \forall y\,Py). \end{array}$$

A simpler companion sequence for A is C_1', C_2' where

$$\begin{array}{lll} C_1' & \text{is} & Pa \Rightarrow \forall y\,Py, \\ C_2' & \text{is} & (Pa \Rightarrow \forall y\,Py) \Rightarrow \exists x\,(Px \Rightarrow \forall y\,Py). \end{array}$$

Exercise 3.2.10. Let A be the logically valid sentence $\exists x\,(Px \Leftrightarrow \forall y\,Py)$. Find a companion sequence $C_1, \ldots, C_n$ for A such that $(C_1 \wedge \cdots \wedge C_n) \Rightarrow A$ is a quasitautology.

Solution. To find a companion sequence for A, we first construct a closed unsigned tableau starting with $\neg\,A$.

$$\begin{array}{ccc} & \neg\,\exists x\,(Px \Leftrightarrow \forall y\,Py) & \\ & \neg\,(Pa \Leftrightarrow \forall y\,Py) & \\ / & & \backslash \\ Pa & & \neg\,Pa \\ \neg\,\forall y\,Py & & \forall y\,Py \\ \neg\,Pb & & Pa \\ \neg\,(Pb \Leftrightarrow \forall y\,Py) & & \\ \begin{array}{cc} / & \backslash \\ Pb & \neg\,Pb \\ \neg\,\forall y\,Py & \forall y\,Py \end{array} & & \end{array}$$

From this tableau, we read off the companion sequence C_1, C_2, C_3, C_4 where

$$\begin{array}{lll} C_1 & \text{is} & (Pa \Leftrightarrow \forall y\,Py) \Rightarrow \exists x\,(Px \Leftrightarrow \forall y\,Py), \\ C_2 & \text{is} & Pb \Rightarrow (\forall y\,Py), \\ C_3 & \text{is} & (Pb \Leftrightarrow \forall y\,Py) \Rightarrow \exists x\,(Px \Leftrightarrow \forall y\,Py), \\ C_4 & \text{is} & (\forall y\,Py) \Rightarrow Pa. \end{array}$$

A simpler companion sequence for A is C_1', C_2', C_3' where

$$\begin{array}{lll} C_1' & \text{is} & Pa \Rightarrow \forall y\, Py, \\ C_2' & \text{is} & (\forall y\, Py) \Rightarrow Pa, \\ C_3' & \text{is} & (Pa \Leftrightarrow \forall y\, Py) \Rightarrow \exists x\, (Px \Leftrightarrow \forall y\, Py). \end{array}$$

3.3 Hilbert-Style Proof Systems

Let L be a language. Recall that V is the set of parameters.

Definition 3.3.1 (the system LH)**.** Our Hilbert-style proof system LH for the predicate calculus is as follows:

1. The objects are L-V-sentences.
2. For each quasitautology A, $\langle A \rangle$ is a rule of inference.
3. $\langle (\forall x\, B) \Rightarrow B[x/a] \rangle$ and $\langle B[x/a] \Rightarrow (\exists x\, B) \rangle$ are rules of inference.
4. $\langle A, A \Rightarrow B, B \rangle$ is a rule of inference.
5. $\langle A \Rightarrow B[x/a], A \Rightarrow (\forall x\, B) \rangle$ and $\langle B[x/a] \Rightarrow A, (\exists x\, B) \Rightarrow A \rangle$ are rules of inference, provided the parameter a does not occur in A or in B.

Schematically, LH consists of:

1. A, where A is any quasitautology
2. (a) $(\forall x\, B) \Rightarrow B[x/a]$ (universal instantiation)
   (b) $B[x/a] \Rightarrow (\exists x\, B)$ (existential instantiation)
3. $\dfrac{A \quad A \Rightarrow B}{B}$ (modus ponens)
4. (a) $\dfrac{A \Rightarrow B[x/a]}{A \Rightarrow (\forall x\, B)}$ (universal generalization)
   (b) $\dfrac{B[x/a] \Rightarrow A}{(\exists x\, B) \Rightarrow A}$ (existential generalization),

   where a does not occur in A, B.

Lemma 3.3.2 (soundness of LH)**.** LH is sound. In other words, for all L-V-sentences A, if A is derivable, then A is logically valid.

Proof. The proof is straightforward by induction on the length of a derivation. The induction step is similar to the proof of Lemma 3.2.4. □

Example 3.3.3. In LH we have the following derivation:

1. $(\forall x\, A) \Rightarrow A[x/a]$ (by universal instantiation)

2. $A[x/a] \Rightarrow (\exists x\, A)$ (by existential instantiation)
3. $((\forall x\, A) \Rightarrow A[x/a]) \Rightarrow ((A[x/a] \Rightarrow (\exists x\, A)) \Rightarrow ((\forall x\, A) \Rightarrow (\exists x\, A)))$

   (This is a quasitautology, obtained from the tautology $(p \Rightarrow q) \Rightarrow ((q \Rightarrow r) \Rightarrow (p \Rightarrow r))$.)
4. $(A[x/a] \Rightarrow (\exists x\, A)) \Rightarrow ((\forall x\, A) \Rightarrow (\exists x\, A))$ (from 1, 3, and modus ponens)
5. $(\forall x\, A) \Rightarrow (\exists x\, A)$ (from 2, 4, and modus ponens)

Thus, by Lemma 3.3.2, $(\forall x\, A) \Rightarrow (\exists x\, A)$ is logically valid.

Example 3.3.4. In LH we have the following derivation:

1. $B[x/a] \Rightarrow (\exists x\, B)$ (by existential instantiation)
2. $(B[x/a] \Rightarrow (\exists x\, B)) \Rightarrow ((A \wedge B)[x/a] \Rightarrow (\exists x\, B))$ (a quasitautology)
3. $(A \wedge B)[x/a] \Rightarrow (\exists x\, B)$ (from 1, 2, and modus ponens)
4. $(\exists x\, (A \wedge B)) \Rightarrow (\exists x\, B)$ (from 3 and existential generalization)

Thus, by Lemma 3.3.2, $(\exists x\, (A \wedge B)) \Rightarrow (\exists x\, B)$ is logically valid.

We now turn to the proof that LH is complete.

Lemma 3.3.5. LH is closed under quasitautological consequence. In other words, if $A_1, \ldots, A_k$ are derivable, and if B is a quasitautological consequence of $A_1, \ldots, A_k$, then B is derivable.

Proof. We are assuming that B is a quasitautological consequence of $A_1, \ldots, A_k$. Thus $A_1 \Rightarrow (A_2 \Rightarrow \cdots \Rightarrow (A_k \Rightarrow B))$ is a quasitautology, hence derivable. We are also assuming that $A_1, \ldots, A_k$ are derivable. Thus we obtain B by k applications of modus ponens. □

Lemma 3.3.6. If C is a companion of A, and if $C \Rightarrow A$ is derivable in LH, then A is derivable in LH.

Proof. First, suppose C is of the form 3.2.3(1), namely $(\forall x\, B) \Rightarrow B[x/a]$. By universal instantiation, C is derivable. In addition, we are assuming that $C \Rightarrow A$ is derivable. Hence, by modus ponens, A is derivable.

Next, suppose C is of the form 3.2.3(2), namely $B[x/a] \Rightarrow (\forall x\, B)$, where a does not occur in A, B. We are assuming that $C \Rightarrow A$ is derivable, i.e., $(B[x/a] \Rightarrow (\forall x\, B)) \Rightarrow A$ is derivable. It follows by Lemma 3.3.5 that both (i) $(\neg A) \Rightarrow B[x/a]$ and (ii) $(\neg A) \Rightarrow (\neg \forall x\, B)$ are derivable. Applying universal generalization to (i), we see that $(\neg A) \Rightarrow (\forall x\, B)$ is derivable. From this plus (ii), it follows by Lemma 3.3.5 that A is derivable.

The other cases, where C is of the form 3.2.3(3) or 3.2.3(4), are handled similarly. □

Theorem 3.3.7 (completeness of LH)**.** LH is sound and complete. In other words, for all L-V-sentences A, A is derivable if and only if A is logically valid.

Proof. The "only if" part is Lemma 3.3.2. For the "if" part, assume that A is logically valid. By Theorem 3.2.7, there exists a companion sequence $C_1, \ldots, C_n$ for A such that $(C_1 \land \cdots \land C_n) \Rightarrow A$ is a quasitautology. Hence $C_n \Rightarrow (C_{n-1} \Rightarrow \cdots \Rightarrow (C_1 \Rightarrow A))$ is a quasitautology, hence derivable. From this and n applications of Lemma 3.3.6, we obtain derivability of A. $\square$

Remark 3.3.8. For convenience in writing proofs, we supplement the rules of LH with

$$\frac{A_1 \ \cdots \ A_k}{B}$$

whenever B is a quasitautological consequence of $A_1, \ldots, A_k$. This is justified by Lemma 3.3.5. We indicate applications of this rule by QT, for quasitautology. Similarly we use UI, EI, UG, EG to indicate universal instantiation, existential instantiation, universal generalization, existential generalization, respectively.

Exercise 3.3.9. Construct a Hilbert-style proof of the sentence

$$(\exists x\, \forall y\, Rxy) \Rightarrow (\forall y\, \exists x\, Rxy).$$

Solution. A proof in LH is

1.	$(\forall y\, Ray) \Rightarrow Rab$	UI
2.	$Rab \Rightarrow (\exists x\, Rxb)$	EI
3.	$(\forall y\, Ray) \Rightarrow (\exists x\, Rxb)$	1,2,QT
4.	$(\forall y\, Ray) \Rightarrow (\forall y\, \exists x\, Rxy)$	3,UG
5.	$(\exists x\, \forall y\, Rxy) \Rightarrow (\forall y\, \exists x\, Rxy)$	4,EG

Exercise 3.3.10. Construct a Hilbert-style proof of the sentence

$$(\forall x\, (Px \land Qx)) \Leftrightarrow ((\forall x\, Px) \land (\forall x\, Qx))$$

Solution. A proof in LH is

1.	$(\forall x\, (Px \land Qx)) \Rightarrow (Pa \land Qa)$	UI
2.	$(\forall x\, (Px \land Qx)) \Rightarrow Pa$	1,QT
3.	$(\forall x\, (Px \land Qx)) \Rightarrow \forall x\, Px$	2,UG
4.	$(\forall x\, (Px \land Qx)) \Rightarrow Qa$	1,QT
5.	$(\forall x\, (Px \land Qx)) \Rightarrow \forall x\, Qx$	4,UG
6.	$(\forall x\, Px) \Rightarrow Pa$	UI
7.	$(\forall x\, Qx) \Rightarrow Qa$	UI
8.	$((\forall x\, Px) \land (\forall x\, Qx)) \Rightarrow (Pa \land Qa)$	6,7,QT
9.	$((\forall x\, Px) \land (\forall x\, Qx)) \Rightarrow \forall x\, (Px \land Qx)$	8,UG
10.	$(\forall x\, (Px \land Qx)) \Leftrightarrow ((\forall x\, Px) \land (\forall x\, Qx))$	3,5,9,QT

Exercise 3.3.11. Construct a Hilbert-style proof of $\neg\, \exists x\, \forall y\, (Eyx \Leftrightarrow \neg\, Eyy)$.

Solution. A proof in LH is

1.	$\forall y\,(Eya \Leftrightarrow \neg\,Eyy) \Rightarrow (Eaa \Leftrightarrow \neg\,Eaa)$	UI
2.	$\neg\,\forall y\,(Eya \Leftrightarrow \neg\,Eyy)$	1,QT
3.	$(\forall y\,(Eya \Leftrightarrow \neg\,Eyy)) \Rightarrow \neg\,\exists x\,\forall y\,(Eyx \Leftrightarrow \neg\,Eyy)$	2,QT
4.	$(\exists x\,\forall y\,(Eyx \Leftrightarrow \neg\,Eyy)) \Rightarrow \neg\,\exists x\,\forall y\,(Eyx \Leftrightarrow \neg\,Eyy)$	3,EG
5.	$\neg\,\exists x\,\forall y\,(Eyx \Leftrightarrow \neg\,Eyy)$	4,QT

Exercise 3.3.12. Construct a Hilbert-style proof of the sentence

$$\neg\,\exists x\,(Sx \wedge \forall y\,(Eyx \Leftrightarrow (Sy \wedge \neg\,Eyy))).$$

Solution. A proof in LH is

1.	$(\forall y\,(Eya \Leftrightarrow (Sy \wedge \neg\,Eyy))) \Rightarrow (Eaa \Leftrightarrow (Sa \wedge \neg\,Eaa))$	UI
2.	$(Sa \wedge (\forall y\,(Eya \Leftrightarrow (Sy \wedge \neg\,Eyy)))) \Rightarrow (Sb \wedge \neg\,Sb)$	1,QT
3.	$(\exists x\,(Sx \wedge (\forall y\,(Eyx \Leftrightarrow (Sy \wedge \neg\,Eyy))))) \Rightarrow (Sb \wedge \neg\,Sb)$	2,EG
4.	$\neg\,\exists x\,(Sx \wedge (\forall y\,(Eyx \Leftrightarrow (Sy \wedge \neg\,Eyy))))$	3,QT

Exercise 3.3.13. Construct a Hilbert-style proof of $\exists x\,(Px \Rightarrow \forall y\,Py)$.

Solution. A proof in LH is

1.	$(Pa \Rightarrow \forall y\,Py) \Rightarrow \exists x\,(Px \Rightarrow \forall y\,Py)$	EI
2.	$(\neg\,\exists x\,(Px \Rightarrow \forall y\,Py)) \Rightarrow Pa$	1,QT
3.	$(\neg\,\exists x\,(Px \Rightarrow \forall y\,Py)) \Rightarrow \forall y\,Py$	2,UG
4.	$\exists x\,(Px \Rightarrow \forall y\,Py)$	1,3,QT

Exercise 3.3.14. Construct a Hilbert-style proof of $\exists x\,(Px \Leftrightarrow \forall y\,Py)$.

Solution. A proof in LH is

1.	$(Pa \Leftrightarrow \forall y\,Py) \Rightarrow \exists x\,(Px \Leftrightarrow \forall y\,Py)$	EI
2.	$(\forall y Py) \Rightarrow Pa$	UI
3.	$(\neg\,\exists x\,(Px \Leftrightarrow \forall y\,Py)) \Rightarrow Pa$	1,2,QT
4.	$(\neg\,\exists x\,(Px \Leftrightarrow \forall y\,Py)) \Rightarrow \forall y\,Py$	3,UG
5.	$\exists x\,(Px \Leftrightarrow \forall y\,Py)$	1,2,4,QT

Exercise 3.3.15. Consider the following proof system LH', which is a "stripped down" version of LH. The objects of LH' are L-V-sentences containing only $\forall$, $\Rightarrow$, $\neg$ (i.e., not containing $\exists$, $\Leftrightarrow$, $\wedge$, $\vee$). The rules of LH' are:

(a) quasitautologies

(b) $(\forall x\,B) \Rightarrow B[x/a]$

(c) $(\forall x\,(A \Rightarrow B)) \Rightarrow (A \Rightarrow \forall x\,B)$

(d) $\dfrac{A \quad A \Rightarrow B}{B}$ (modus ponens)

(e) $\dfrac{B[x/a]}{\forall x\, B}$ (generalization), where a does not occur in B.

Show that LH' is sound and complete.

Solution. Soundness is proved just as for LH.

Just as for the full tableau method, we can prove soundness and completeness of the restricted tableau method with $\forall$, $\Rightarrow$, $\neg$, and from this we obtain the restricted Companion Theorem. There are now only two kinds of companions, the ones involving $\forall$. It remains to prove the following lemma: If C is a companion of A, and if $C \Rightarrow A$ is derivable in LH', then A is derivable in LH'.

Consider a companion of the form $B[x/a] \Rightarrow (\forall x\, B)$. Assume that

$$(B[x/a] \Rightarrow (\forall x\, B)) \Rightarrow A$$

is derivable in LH', where a does not occur in A, B. It follows quasitautologically that both (1) $(\neg A) \Rightarrow B[x/a]$ and (2) $(\neg A) \Rightarrow \neg \forall x\, B$ are derivable in LH'. From (1) and the generalization rule (e) of LH', we see that $\forall x\, ((\neg A) \Rightarrow B)$ is derivable in LH'. Also, by rule (c) of LH',

$$(\forall x\, ((\neg A) \Rightarrow B)) \Rightarrow ((\neg A) \Rightarrow \forall x\, B)$$

is derivable in LH'. Hence, by modus ponens, $(\neg A) \Rightarrow \forall x\, B$ is derivable in LH'. It follows quasitautologically from this and (2) that A is derivable in LH'. This completes the proof.

Exercise 3.3.16.

1. Let S be a set of L-sentences. Consider a proof system $LH(S)$ consisting of LH with additional rules of inference $\langle A \rangle$, $A \in S$. Show that an L-V-sentence B is derivable in $LH(S)$ if and only if B is a logical consequence of S.

2. Indicate the modifications needed when S is a set of L-V-sentences.

Solution.

1. By induction on the length of derivations, it is straightforward to prove that each sentence derivable in $LH(S)$ is a logical consequence of S. The assumption that S is a set of L-sentences (not L-V-sentences) is used in the inductive steps corresponding to rules 4(a) and 4(b), universal and existential generalization, because we need to know that the parameter a does not occur in S.

   Conversely, assume B is a logical consequence of S. By the Compactness Theorem, it follows that B is a logical consequence of a finite subset of S, say $A_1, \ldots, A_n$. Hence $(A_1 \wedge \cdots \wedge A_n) \Rightarrow B$ is logically valid. Hence, by completeness of LH, $(A_1 \wedge \cdots \wedge A_n) \Rightarrow B$ is derivable in LH. Since $LH(S)$ includes LH, we have that

$$(A_1 \wedge \cdots \wedge A_n) \Rightarrow B$$

is derivable in $LH(S)$. But $A_1, \ldots, A_n$ are derivable in $LH(S)$. It follows quasitautologically that B is derivable in $LH(S)$. This completes the proof.

2. If S is a set of L-V-sentences, we modify our system as follows. Let V' be a countably infinite set of new parameters, disjoint from V. Define $LH(S)$ as before, but allowing parameters from $V \cup V'$. The objects are L-$V \cup V'$-sentences. In rules 4(a) and 4(b), one must impose the restriction that a does not occur in A, B, S. With this modification, everything goes through as before.

Notation 3.3.17. We write $S \vdash B$ to indicate that B is derivable in $LH(S)$.

Exercise 3.3.18. Let S be an infinite set of L-sentences, and let B be an L-sentence. Prove that $S \models B$ (i.e., B is true in all L-structures satisfying S) if and only if there exists a finite set of L-sentences $A_1, \ldots, A_k \in S$ such that $A_1, \ldots, A_k \vdash B$ (i.e., B is provable from $A_1, \ldots, A_k$).

Solution. The "if" part follows from the soundness of our proof system. For the "only if" part, assume that $S \models B$, i.e., $S \cup \{\neg B\}$ is not satisfiable. It follows by the Compactness Theorem that there exists a finite set $\{A_1, \ldots, A_k\} \subset S$ such that $A_1, \ldots, A_k, \neg B$ is not satisfiable. Thus B is a logical consequence of $A_1, \ldots, A_k$. It follows by completeness of our proof system that $A_1, \ldots, A_k \vdash B$.

3.4 Gentzen-Style Proof Systems

Throughout this section, let L be a language. As usual, V is the set of parameters.

Before presenting our Gentzen-style proof system for L, we first discuss the block tableau method, a trivial variant of the signed tableau method.

Definition 3.4.1. A *block* is a finite set of signed L-V-sentences. A block is said to be *closed* if it contains T A and F A for some L-V-sentence A.

Notation 3.4.2. If S is a block and X is a signed L-V-sentence, we write S, X instead of $S \cup \{X\}$, etc.

Definition 3.4.3. A *block tableau* is a rooted dyadic tree where each node carries a block. A block tableau is said to be *closed* if each of its end nodes is closed. Given a block S, a *block tableau starting with* S is a block tableau generated from S by means of block tableau rules. The *block tableau rules* are obtained from the signed tableau rules (pages 13 and 31) as follows. Corresponding to signed tableau rules of the form

$$
\begin{array}{c}
\vdots \\ X \\ \vdots \\ | \\ Y
\end{array}
\qquad
\begin{array}{c}
\vdots \\ X \\ \vdots \\ | \\ Y_1 \\ Y_2
\end{array}
\qquad
\begin{array}{c}
\vdots \\ X \\ \vdots \\ / \ \backslash \\ Y \quad Z
\end{array}
\qquad
\begin{array}{c}
\vdots \\ X \\ \vdots \\ / \ \backslash \\ Y_1 \quad Z_1 \\ Y_2 \quad Z_2
\end{array}
$$

we have block tableau rules

$$
\begin{array}{c}
S, X \\ | \\ S, X, Y
\end{array}
\qquad
\begin{array}{c}
S, X \\ | \\ S, X, Y_1, Y_2
\end{array}
\qquad
\begin{array}{c}
S, X \\ / \ \backslash \\ S, X, Y \quad S, X, Z
\end{array}
\qquad
\begin{array}{c}
S, X \\ / \ \backslash \\ S, X, Y_1, Y_2 \quad S, X, Z_1, Z_2
\end{array}
$$

respectively.

For example, we have the following block tableau rules:

$$
\begin{array}{c}
S, \mathrm{T}\, A \wedge B \\ | \\ S, \mathrm{T}\, A \wedge B, \mathrm{T}\, A, \mathrm{T}\, B
\end{array}
\qquad
\begin{array}{c}
S, \mathrm{F}\, A \wedge B \\ / \ \backslash \\ S, \mathrm{F}\, A \wedge B, \mathrm{F}\, A \quad S, \mathrm{F}\, A \wedge B, \mathrm{F}\, B
\end{array}
$$

$$
\begin{array}{c}
S, \mathrm{T}\, A \Rightarrow B \\ / \ \backslash \\ S, \mathrm{T}\, A \Rightarrow B, \mathrm{F} A \quad S, \mathrm{T}\, A \Rightarrow B, \mathrm{T} B
\end{array}
\qquad
\begin{array}{c}
S, \mathrm{F}\, A \Rightarrow B \\ | \\ S, \mathrm{F}\, A \Rightarrow B, \mathrm{T}\, A, \mathrm{F}\, B
\end{array}
$$

$$
\begin{array}{c}
S, \mathrm{T}\, \forall x\, A \\ | \\ S, \mathrm{T}\, \forall x\, A, \mathrm{T}\, A[x/a]
\end{array}
\qquad
\begin{array}{c}
S, \mathrm{F}\, \forall x\, A \\ | \\ S, \mathrm{F}\, \forall x\, A, \mathrm{F}\, A[x/a] \\ \text{where } a \text{ is new.}
\end{array}
$$

Example 3.4.4. We exhibit a closed block tableau demonstrating that $\exists x\, A$ is a logical consequence of $\forall x\, A$.

$$
\begin{array}{c}
\mathrm{T}\, \forall x\, A,\ \mathrm{F}\, \exists x\, A \\ | \\ \mathrm{T}\, \forall x\, A,\ \mathrm{F}\, \exists x\, A,\ \mathrm{T}\, A[x/a] \\ | \\ \mathrm{T}\, \forall x\, A,\ \mathrm{F}\, \exists x\, A,\ \mathrm{T}\, A[x/a],\ \mathrm{F}\, A[x/a]
\end{array}
$$

This block tableau is of course similar to the signed tableau

$$
\begin{array}{c}
\mathrm{T}\, \forall x\, A \\ \mathrm{F}\, \exists x\, A \\ \mathrm{T}\, A[x/a] \\ \mathrm{F}\, A[x/a]
\end{array}
$$

which demonstrates the same thing.

We now define our Gentzen-style system, LG.

Definition 3.4.5. A *sequent* is an expression of the form $\Gamma \to \Delta$ where Γ and Δ are finite sets of L-V-sentences. If $S = \mathrm{T}\,A_1, \ldots, \mathrm{T}\,A_m, \mathrm{F}\,B_1, \ldots, \mathrm{F}\,B_n$ is a block, let $|S|$ be the sequent $A_1, \ldots, A_m \to B_1, \ldots, B_n$. This gives a one-to-one correspondence between blocks and sequents.

Definition 3.4.6 (the system LG)**.** Our Gentzen-style proof system LG for the predicate calculus is as follows.

1. The objects of LG are sequents.[2]

2. For each closed block S, we have a rule of inference $\langle |S| \rangle$. In other words, for all finite sets of L-V-sentences Γ and Δ and all L-V-sentences A, we assume the sequent $\Gamma, A \to A, \Delta$.

3. For each non-branching block tableau rule

$$\begin{array}{c} S \\ | \\ S' \end{array}$$

   we have a rule of inference $\langle |S'|, |S| \rangle$, i.e., $\dfrac{|S'|}{|S|}$.

4. For each branching block tableau rule

$$\begin{array}{ccc} & S & \\ / & & \backslash \\ S' & & S'' \end{array}$$

   we have a rule of inference $\langle |S'|, |S''|, |S| \rangle$, i.e., $\dfrac{|S'| \quad |S''|}{|S|}$.

Thus LG includes the following rules of inference:

[2]For this reason, LG is sometimes called a *sequent calculus*.

$$\overline{\Gamma, A \to A, \Delta}$$

$$\frac{\Gamma, \neg A \to A, \Delta}{\Gamma, \neg A \to \Delta} \qquad \frac{\Gamma, A \to \neg A, \Delta}{\Gamma \to \neg A, \Delta}$$

$$\frac{\Gamma, A \wedge B, A, B \to \Delta}{\Gamma, A \wedge B \to \Delta} \qquad \frac{\Gamma \to A \wedge B, A, \Delta \quad \Gamma \to A \wedge B, B, \Delta}{\Gamma \to A \wedge B, \Delta}$$

$$\frac{\Gamma, A \Rightarrow B \to A, \Delta \quad \Gamma, A \Rightarrow B, B \to \Delta}{\Gamma, A \Rightarrow B \to \Delta} \qquad \frac{\Gamma, A \to A \Rightarrow B, B, \Delta}{\Gamma \to A \Rightarrow B, \Delta}$$

$$\frac{\Gamma, \forall x\, A, A[x/a] \to \Delta}{\Gamma, \forall x\, A \to \Delta} \qquad \frac{\Gamma \to \forall x\, A, A[x/a], \Delta}{\Gamma \to \forall x\, A, \Delta}$$

where a does not occur in the conclusion.

Exercise 3.4.7. Explicitly display the remaining inference rules of LG.

Definition 3.4.8. A sequent $A_1, \ldots, A_m \to B_1, \ldots, B_n$ is said to be *logically valid* if and only if the L-V-sentence

$$(A_1 \wedge \cdots \wedge A_m) \Rightarrow (B_1 \vee \cdots \vee B_n)$$

is logically valid.[3]

Theorem 3.4.9 (soundness and completeness of LG)**.** *LG is sound and complete. In other words, a sequent $\Gamma \to \Delta$ is logically valid if and only if it is derivable in LG. In particular, an L-V-sentence A is logically valid if and only if the sequent*

$$\to A$$

is derivable in LG.

Proof. Note that the sequent $A_1, \ldots, A_m \to B_1, \ldots, B_n$ is logically valid if and only if the block $\mathrm{T}\,A_1, \ldots, \mathrm{T}\,A_m, \mathrm{F}\,B_1, \ldots, \mathrm{F}\,B_n$ is not satisfiable. Thus, soundness and completeness of LG is equivalent to soundness and completeness of the block tableau method. The latter is in turn easily seen to be equivalent to soundness and completeness of the signed tableau method, as presented in Theorems 2.3.13 and 2.5.5. ∎

[3] In particular, the sequents $A_1, \ldots, A_m \to$ and $\to B_1, \ldots, B_n$ are said to be logically valid if and only if the L-V-sentences $\neg(A_1 \wedge \cdots \wedge A_m)$ and $B_1 \vee \cdots \vee B_n$ are logically valid, respectively. The empty sequent $\to$ is deemed not logically valid.

Exercise 3.4.10. Construct a Gentzen-style proof of the sequent

$$\exists x\,\forall y\,Rxy \to \forall y\,\exists x\,Rxy.$$

Solution. A proof in LG is

1. $\exists x\,\forall y\,Rxy, \forall y\,Ray, Rab \to Rab, \exists x\,Rxb, \forall y\,\exists x\,Rxy$
2. $\exists x\,\forall y\,Rxy, \forall y\,Ray, Rab \to \exists x\,Rxb, \forall y\,\exists x\,Rxy$
3. $\exists x\,\forall y\,Rxy, \forall y\,Ray \to \exists x\,Rxb, \forall y\,\exists x\,Rxy$
4. $\exists x\,\forall y\,Rxy \to \exists x\,Rxb, \forall y\,\exists x\,Rxy$
5. $\exists x\,\forall y\,Rxy \to \forall y\,\exists x\,Rxy$

Definition 3.4.11. In order to simplify the writing of Gentzen-style proofs, let LG^+ be LG augmented with the so-called *weakening rules* or *padding rules*:

$$\frac{\Gamma \to \Delta}{\Gamma, A \to \Delta} \qquad\qquad \frac{\Gamma \to \Delta}{\Gamma \to A, \Delta}$$

where A is an L-V-sentence. Clearly LG^+ is sound and complete.

Remark 3.4.12. Any proof in LG is a proof in LG^+, and any proof in LG^+ may be straightforwardly "padded out" to a proof in LG. Thus LG^+ differs only slightly from LG. However, proofs in LG^+ are easier. For example, patterned on the above proof in LG, we have the following proof in LG^+:

1. $Rab \to Rab$
1.5. $\forall y\,Ray, Rab \to Rab$
2. $\forall y\,Ray \to Rab$
2.5. $\forall y\,Ray \to Rab, \exists x\,Rxb$
3. $\forall y\,Ray \to \exists x\,Rxb$
3.5. $\exists x\,\forall y\,Rxy, \forall y\,Ray \to \exists x\,Rxb$
4. $\exists x\,\forall y\,Rxy \to \exists x\,Rxb$
4.5. $\exists x\,\forall y\,Rxy \to \exists x\,Rxb, \forall y\,\exists x\,Rxy$
5. $\exists x\,\forall y\,Rxy \to \forall y\,\exists x\,Rxy$

or, omitting the applications of the padding rules,

1. $Rab \to Rab$
2. $\forall y\,Ray \to Rab$
3. $\forall y\,Ray \to \exists x\,Rxb$
4. $\exists x\,\forall y\,Rxy \to \exists x\,Rxb$
5. $\exists x\,\forall y\,Rxy \to \forall y\,\exists x\,Rxy$

Exercise 3.4.13. Construct a Gentzen-style proof of the sequent

$$\to (\exists x\,\forall y\,Rxy) \Rightarrow (\forall y\,\exists x\,Rxy).$$

Solution. A proof in LG^+ consists of the previous proof followed by

5.5. $\exists x\,\forall y\,Rxy \to \forall y\,\exists x\,Rxy, (\exists x\,\forall y\,Rxy) \Rightarrow (\forall y\,\exists x\,Rxy)$
6. $\to (\exists x\,\forall y\,Rxy) \Rightarrow (\forall y\,\exists x\,Rxy)$

Exercise 3.4.14. Construct a Gentzen-style proof of the sequent

$$\to \neg \exists x \, (Sx \land \forall y \, (Eyx \Leftrightarrow (Sy \land \neg Eyy))).$$

Solution. A proof in LG^+ with padding rules omitted is

1.	$Eaa \to Eaa$	axiom
2.	$Eaa, \neg Eaa \to$	from 1
3.	$Eaa, Sa \land \neg Eaa \to$	from 2
4.	$Sa \to Sa$	axiom
5.	$\to Eaa, \neg Eaa$	from 1
6.	$Sa \to Eaa, Sa \land \neg Eaa$	from 4 and 5
7.	$Sa, Eaa \Leftrightarrow (Sa \land \neg Eaa) \to$	from 4 and 6
8.	$Sa, \forall y \, (Eya \Leftrightarrow (Sy \land \neg Eyy)) \to$	from 7
9.	$Sa \land \forall y \, (Eya \Leftrightarrow (Sy \land \neg Eyy)) \to$	from 8
10.	$\exists x \, (Sx \land \forall y \, (Eyx \Leftrightarrow (Sy \land \neg Eyy)) \to$	from 9
11.	$\to \neg \exists x \, (Sx \land \forall y \, (Eyx \Leftrightarrow (Sy \land \neg Eyy))$	from 10

Exercise 3.4.15. Construct a Gentzen-style proof of $\exists x \, (Px \Rightarrow \forall y \, Py)$.

Solution. A proof in LG^+ with padding rules omitted is

1.	$Pa \to Pa$	axiom
2.	$\to Pa, Pa \Rightarrow \forall y \, Py$	from 1
3.	$\to Pa, \exists x \, (Px \Rightarrow \forall y \, Py)$	from 2
4.	$\to \forall y \, Py, \exists x \, (Px \Rightarrow \forall y \, Py)$	from 3
5.	$\to Pa \Rightarrow \forall y \, Py, \exists x \, (Px \Rightarrow \forall y \, Py)$	from 4
6.	$\to \exists x \, (Px \Rightarrow \forall y \, Py)$	from 5

Exercise 3.4.16. Construct a Gentzen-style proof of $\exists x \, (Px \Leftrightarrow \forall y \, Py)$.

Solution. A proof in LG^+ with padding rules omitted is

1.	$Pa \to Pa, \forall y \, Py$	axiom
2.	$\forall y \, Py \to Pa, \forall y \, Py$	axiom
3.	$\to Pa \Leftrightarrow \forall y \, Py, Pa, \forall y \, Py$	from 1 and 2 via $\to\Leftrightarrow$
4.	$\to Pa \Leftrightarrow \forall y \, Py, \forall y \, Py$	from 3 via $\to\forall$
5.	$Pb \to Pb$	axiom
6.	$\forall y \, Py \to Pb$	from 5 via $\forall\to$
7.	$\to Pa \Leftrightarrow \forall y \, Py, Pb \Leftrightarrow \forall y \, Py$	from 4 and 6 via $\to\Leftrightarrow$
8.	$\to Pa \Leftrightarrow \forall y \, Py, \exists x \, (Px \Leftrightarrow \forall y \, Py)$	from 7 via $\to\exists$
9.	$\to \exists x \, (Px \Leftrightarrow \forall y \, Py)$	from 8 via $\to\exists$

The above proof was derived from the tableau in Exercise 3.2.10.

Exercise 3.4.17. Let LG(atomic) be a variant of LG in which $\Gamma, A \to A, \Delta$ is assumed only for atomic L-V-sentences A. Show that LG(atomic) is sound and complete. (Hint: Use the result of Exercise 2.5.7.)

Exercise 3.4.18.

1. The *modified block tableau rules* are a variant of the block tableau rules of Definition 3.4.3, replacing each non-branching rule of the form

$$\begin{array}{c} S, X \\ | \\ S, X, Y_1, Y_2 \end{array}$$

by a pair of rules

$$\begin{array}{c} S, X \\ | \\ S, X, Y_1 \end{array} \qquad\qquad \begin{array}{c} S, X \\ | \\ S, X, Y_2 \end{array}$$

Show that the modified block tableau rules are sound and complete.

2. Let LG' be the variant of LG corresponding to the modified block tableau rules. Write out all the rules of LG' explicitly. Show that LG' is sound and complete.

3.5 The Interpolation Theorem

As usual, let L be a language and let V be the set of parameters.

Theorem 3.5.1 (the Interpolation Theorem)**.** Let A and B be L-V-sentences. If $A \Rightarrow B$ is logically valid, we can find an L-V-sentence I such that:

1. $A \Rightarrow I$ and $I \Rightarrow B$ are logically valid.

2. Each predicate and parameter occurring in I occurs in both A and B.

Such an I is called an *interpolant* for $A \Rightarrow B$. We indicate this by writing $A \overset{I}{\Rightarrow} B$.

Remark 3.5.2. If A and B have no predicates in common, then obviously the theorem is incorrect as stated, because all L-V-sentences necessarily contain at least one predicate. In this case, we modify the conclusion of the theorem to say that at least one of $\neg A$ and B is logically valid.[4] The conclusion is obvious in this case.

In order to prove the Interpolation Theorem, we introduce a "symmetric" variant of LG, wherein sentences do not move from one side of $\rightarrow$ to the other.

Definition 3.5.3. A *signed sequent* is an expression of the form $M \rightarrow N$ where M and N are finite sets of signed L-V-sentences. A *variant* of $M \rightarrow N$ is a signed sequent obtained from $M \rightarrow N$ by transferring sentences from one side of $\rightarrow$ to the other, changing signs. In particular, $M, X \rightarrow N$ and $M \rightarrow \overline{X}, N$ are variants of each other, where we use an overline to denote conjugation, i.e., $\overline{\mathrm{T}\,A} = \mathrm{F}\,A$, $\overline{\mathrm{F}\,A} = \mathrm{T}\,A$.

[4] This amounts to saying that at least one of the truth values T and F is an interpolant for $A \Rightarrow B$.

Definition 3.5.4. Let

$$C_1, \ldots, C_m \to D_1, \ldots, D_n$$

be an unsigned sequent[5]. A *signed variant* of $C_1, \ldots, C_m \to D_1, \ldots, D_n$ is any variant of the signed sequent

$$\mathrm{T}\, C_1, \ldots, \mathrm{T}\, C_m \to \mathrm{T}\, D_1, \ldots, \mathrm{T}\, D_n\,.$$

Note that each signed sequent is a signed variant of one and only one unsigned sequent. We define a signed sequent to be *logically valid* if and only if the corresponding unsigned sequent is logically valid.

Definition 3.5.5. LG(symmetric) is the following proof system.

1. The objects are signed sequents.

2. We have

$$\overline{M, X \to X, N}$$

and

$$\overline{M, X, \overline{X} \to N}$$

and

$$\overline{M \to X, \overline{X}, N}$$

for all X.

3. For each signed tableau rule of the form

$$\begin{array}{cccc} \vdots & \vdots & \vdots & \vdots \\ X & X & X & X \\ \vdots & \vdots & \vdots & \vdots \\ | & | & / \ \ \backslash & / \ \ \backslash \\ Y & Y_1 & Y \quad Z & Y_1 \quad Z_1 \\ & Y_2 & & Y_2 \quad Z_2 \end{array}$$

we have a corresponding pair of signed sequent rules

[5] An *unsigned sequent* is just what we have previously called a sequent.

$$\frac{M, X, Y \to N}{M, X \to N} \qquad \frac{M \to \overline{X}, \overline{Y}, N}{M \to \overline{X}, N}$$

$$\frac{M, X, Y_1, Y_2 \to N}{M, X \to N} \qquad \frac{M \to \overline{X}, \overline{Y_1}, \overline{Y_2}, N}{M \to \overline{X}, N}$$

$$\frac{M, X, Y \to N \quad M, X, Z \to N}{M, X \to N} \qquad \frac{M \to \overline{X}, \overline{Y}, N \quad M \to \overline{X}, \overline{Z}, N}{M \to \overline{X}, N}$$

$$\frac{M, X, Y_1, Y_2 \to N \quad M, X, Z_1, Z_2 \to N}{M, X \to N} \qquad \frac{M \to \overline{X}, \overline{Y_1}, \overline{Y_2}, N \quad M \to \overline{X}, \overline{Z_1}, \overline{Z_2}, N}{M \to \overline{X}, N}$$

respectively.

Lemma 3.5.6. An unsigned sequent is derivable in LG if and only if all of its signed variants are derivable in LG(symmetric).

Proof. The proof is by induction on the length of derivations in LG. The base step consists of noting that all signed variants of $\Gamma, A \to A, \Delta$ are of the form $M, X \to X, N$ or $M, X, \overline{X} \to N$ or $M \to X, \overline{X}, N$, hence derivable in LG(symmetric). The inductive step consists of checking that, for each rule of inference of LG, if all signed variants of the premises are derivable in LG(symmetric), then so are all signed variants of the conclusion. This is straightforward. □

Theorem 3.5.7. LG(symmetric) is sound and complete. In other words, a signed sequent is logically valid if and only if it is derivable in LG(symmetric). In particular, an L-V-sentence $A \Rightarrow B$ is logically valid if and only if the signed sequent $\mathrm{T}\,A \to \mathrm{T}\,B$ is derivable in LG(symmetric).

Proof. Soundness and completeness of LG(symmetric) follows from Theorem 3.4.9, soundness and completeness of LG, using Lemma 3.5.6. □

We now prove the Interpolation Theorem.

Definition 3.5.8. Let $M \to N$ be a signed sequent. An *interpolant* for $M \to N$ is an L-V-sentence I such that the signed sequents $M \to \mathrm{T}\,I$ and $\mathrm{T}\,I \to N$ are logically valid, and all predicates and parameters occurring in I occur in both M and N.[6] We indicate this by writing $M \xrightarrow{I} N$.

In order to prove the Interpolation Theorem, it suffices by Theorem 3.5.7 to prove that every signed sequent derivable in LG(symmetric) has an interpolant. We prove this by induction on the length of derivations.

[6]In the special case when M and N have no predicates in common, we require instead that at least one of the signed sequents $M \to$ and $\to N$ be logically valid. This amounts to requiring that at least one of T, F be an interpolant for $M \to N$.

For the base step, we note that X is an interpolant for $M, X \to X, N$, and that $M, X, \overline{X} \to$ and $\to X, \overline{X}, N$ are logically valid. Thus we have $M, X \xrightarrow{X} X, N$ and $M, X, \overline{X} \xrightarrow{\mathrm{F}} N$ and $M \xrightarrow{\mathrm{T}} X, \overline{X}, N$.

For the induction step we show that, for each rule of LG(symmetric), given interpolants for the premises of the rule, we can find an interpolant for the conclusion. We present some representative special cases.

$$\frac{M, \mathrm{T}\, A \wedge B, \mathrm{T}\, A, \mathrm{T}\, B \xrightarrow{I} N}{M, \mathrm{T}\, A \wedge B \xrightarrow{I} N} \qquad \frac{M \xrightarrow{I} \mathrm{F}\, A \wedge B, \mathrm{F}\, A, \mathrm{F} B, N}{M \xrightarrow{I} \mathrm{F}\, A \wedge B, N}$$

$$\frac{M, \mathrm{F}\, A \wedge B, \mathrm{F}\, A \xrightarrow{I} N \qquad M, \mathrm{F}\, A \wedge B, \mathrm{F}\, B \xrightarrow{J} N}{M, \mathrm{F}\, A \wedge B \xrightarrow{I \vee J} N}$$

$$\frac{M \xrightarrow{I} \mathrm{T}\, A \wedge B, \mathrm{T}\, A, N \qquad M \xrightarrow{J} \mathrm{T}\, A \wedge B, \mathrm{T}\, B, N}{M \xrightarrow{I \wedge J} \mathrm{T}\, A \wedge B, N}$$

$$\frac{M, \mathrm{T} \neg A, \mathrm{F}\, A \xrightarrow{I} N}{M, \mathrm{T} \neg A \xrightarrow{I} N} \qquad \frac{M \xrightarrow{I} \mathrm{F} \neg A, \mathrm{T}\, A, N}{M \xrightarrow{I} \mathrm{F} \neg A, N}$$

$$\frac{M, \mathrm{F} \neg A, \mathrm{T}\, A \xrightarrow{I} N}{M, \mathrm{F} \neg A \xrightarrow{I} N} \qquad \frac{M \xrightarrow{I} \mathrm{T} \neg A, \mathrm{F}\, A, N}{M \xrightarrow{I} \mathrm{T} \neg A, N}$$

$$\frac{M, \mathrm{F} \forall x\, A, \mathrm{F}\, A[x/a] \xrightarrow{I} N}{M, \mathrm{F} \forall x\, A \xrightarrow{I} N}$$

where a does not occur in the conclusion.

$$\frac{M \xrightarrow{I} \mathrm{T} \forall x\, A, \mathrm{T}\, A[x/a], N}{M \xrightarrow{I} \mathrm{T} \forall x\, A, N}$$

where a does not occur in the conclusion.

$$\frac{M, \mathrm{T} \forall x\, A, \mathrm{T}\, A[x/a] \xrightarrow{I} N}{M, \mathrm{T} \forall x\, A \xrightarrow{K} N}$$

where $K = I$ if a occurs in $M, \mathrm{T} \forall x\, A$, otherwise $K = \forall z\, I[a/z]$ where z is a new variable.

$$\frac{M \xrightarrow{I} \mathrm{F} \forall x\, A, \mathrm{F}\, A[x/a], N}{M \xrightarrow{K} \mathrm{F} \forall x\, A, N}$$

where $K = I$ if a occurs in $\mathrm{F} \forall x\, A, N$, otherwise $K = \exists z\, I[a/z]$ where z is a new variable.

This completes the proof.

Example 3.5.9. We give an example illustrating the Interpolation Theorem. Let n be a large positive integer, say $n = 1000$. Let A_n say that the universe consists of the vertices of a simple, undirected graph with a clique of size n. Let B_n say that the graph is not $(n-1)$-colorable. Both A_n and B_n contain a predicate R denoting adjacency in the graph. A_n contains a unary predicate Q denoting a clique. B_n contains a binary predicate E saying that two vertices get the same color.

A_n is:

> R and G are irreflexive relations on the universe, R is symmetric, G is transitive, $\forall x\, \forall y\, ((Qx \wedge Qy \wedge Gxy) \Rightarrow Rxy)$, and there exist $x_1, \ldots, x_n$ such that $Qx_1 \wedge \cdots \wedge Qx_n \wedge Gx_1x_2 \wedge \cdots \wedge Gx_{n-1}x_n$.

B_n is the negation of:

> E is an equivalence relation on the universe, $\forall x\, \forall y\, (Rxy \Rightarrow \neg\, Exy)$, and there exist $x_1, \ldots, x_{n-1}$ such that $\forall y\, (Ex_1y \vee \cdots \vee Ex_{n-1}y)$.

Clearly $A_n \Rightarrow B_n$ is logically valid. Note that the lengths of A_n and B_n are $O(n)$, i.e., proportional to n. The obvious interpolant I_n says there exists a clique of size n, i.e., there exist $x_1, \ldots, x_n$ such that $Rx_1x_2 \wedge Rx_1x_3 \wedge \cdots \wedge Rx_{n-1}x_n$. Note that the length of I_n is $O(n^2)$, i.e., proportional to n^2. It appears that there is no interpolant of length $O(n)$.

Chapter 4

Extensions of Predicate Calculus

In this chapter we consider various extensions of the predicate calculus. These extensions may be regarded as inessential features or "bells and whistles" which are introduced solely in order to make the predicate calculus more user-friendly.

4.1 Predicate Calculus with Identity

Definition 4.1.1. A *language with identity* consists of a language L with a particular binary predicate, I, designated as the *identity predicate*.

Definition 4.1.2. Let L be a language with identity. The *identity axioms for L* are the following sentences:

1. $\forall x\, Ixx$ (reflexivity)
2. $\forall x\, \forall y\, (Ixy \Leftrightarrow Iyx)$ (symmetry)
3. $\forall x\, \forall y\, \forall z\, ((Ixy \wedge Iyz) \Rightarrow Ixz)$ (transitivity)
4. For each n-ary predicate P of L, we have an axiom
   $\forall x_1 \cdots \forall x_n\, \forall y_1 \cdots \forall y_n\, ((Ix_1y_1 \wedge \cdots \wedge Ix_ny_n) \Rightarrow (Px_1 \cdots x_n \Leftrightarrow Py_1 \cdots y_n))$
   (congruence).

Exercise 4.1.3. Show that the identity predicate is unique in the following sense. If L contains two identity predicates I_1 and I_2, then $\forall x\, \forall y\, (I_1xy \Leftrightarrow I_2xy)$ is a logical consequence of the identity axioms for I_1 and I_2.

Let L be a language with identity.

Definition 4.1.4. An L-structure M is said to be *normal* if the identity predicate denotes the identity relation, i.e., $I_M = \{\langle a, a \rangle \mid a \in U_M\}$.

Note that any normal L-structure automatically satisfies the identity axioms for L. Conversely, we have:

Theorem 4.1.5. Let M be an L-structure satisfying the identity axioms for L. For each $a \in U_M$ put $\overline{a} = \{b \in U_M \mid v_M(Iab) = \mathrm{T}\}$. Then we have a normal L-structure $\overline{M}$ and an onto mapping $\phi : U_M \to U_{\overline{M}}$ as in Theorem 2.7.3, defined by putting $U_{\overline{M}} = \{\overline{a} \mid a \in U_M\}$, and $P_{\overline{M}} = \{\langle \overline{a}_1, \ldots, \overline{a}_n \rangle \mid \langle a_1, \ldots, a_n \rangle \in P_M\}$ for all n-ary predicates P.

Proof. This is straightforward, using the fact that I_M is a congruence with respect to each of the relations P_M, $P \in L$. □

Theorem 4.1.6. If M is an L-structure satisfying the identity axioms for L, then we have a normal L-structure $\overline{M}$ satisfying the same sentences as M.

Proof. This is immediate from Theorems 4.1.5 and 2.7.3. □

Let S be a set of L-sentences.

Definition 4.1.7. S is *normally satisfiable* if there exists a normal L-structure which satisfies S.

Corollary 4.1.8. S is normally satisfiable if and only if

$$S \cup \{\text{identity axioms for } L\}$$

is satisfiable.

We also have the Compactness Theorem for normal satisfiability:

Corollary 4.1.9. S is normally satisfiable if and only if each finite subset of S is normally satisfiable.

Proof. This is immediate from Corollary 4.1.8 plus the Compactness Theorem for predicate calculus without identity (Theorems 2.6.1 and 2.6.2), applied to the set $S \cup \{\text{identity axioms for } L\}$. □

Regarding normal satisfiability in particular domains, we have:

Example 4.1.10. Given a positive integer n, we exhibit a sentence E_n which is normally satisfiable in domains of cardinality n but not in domains of any other cardinality. The sentence

$$\exists x_1 \cdots \exists x_n \, (\forall y \, (Ix_1 y \vee \cdots \vee Ix_n y) \wedge \neg \, (Ix_1 x_2 \vee Ix_1 x_3 \vee \cdots \vee Ix_{n-1} x_n))$$

has this property. Intuitively, E_n says that there exist exactly n things.

Exercise 4.1.11. Let $M = (U_M, f_M, g_M, I_M)$ where $U_M = \{0, 1, 2, 3, 4\}$, I_M is the identity relation on U_M, and f_M, g_M are the binary operations of addition and multiplication modulo 5. Thus M is essentially just the ring of integers modulo 5. Let L be the language consisting of f, g, I. Note that M is a normal L-structure.

Write an L-sentence A such that for all normal L-structures M', M' satisfies A if and only if M' is isomorphic to M.

Solution. A brute force solution is to let A be $\exists x_0\,\exists x_1\,\exists x_2\,\exists x_3\,\exists x_4\,B$, where B is the conjunction of $\{\forall y\,(Ix_0y \vee Ix_1y \vee Ix_2y \vee Ix_3y \vee Ix_4y)\} \cup \{\neg\, Ix_ix_j : i \neq j\} \cup \{Ifx_ix_jx_k \mid i+j = k \bmod 5\} \cup \{Igx_ix_jx_k \mid ij = k \bmod 5\}$ with i,j,k ranging over $0,1,2,3,4$.

Another solution is to let A be a sentence describing a field consisting of 5 elements. Namely, let A be the conjunction of the field axioms plus "there exist exactly 5 things". We are using the algebraic fact that, up to isomorphism, there is exactly one field of 5 elements.

Exercise 4.1.12. Let L be a finite language with identity, and let M be a finite normal L-structure. Construct an L-sentence A such that, for all normal L-structures M', $M' \models A$ if and only if M' is isomorphic to M.

Solution. Let $a_1,\ldots,a_k$ be the elements of U_M, and let $P,\ldots,Q$ be the predicates of L. As A we may take

$$\exists x_1 \cdots \exists x_k\,(D_P \wedge \cdots \wedge D_Q \wedge \forall y\,(Ix_1y \vee \cdots \vee Ix_ky))$$

where for each n-ary predicate P of L, D_P is the conjunction of $Px_{i_1}\cdots x_{i_n}$ for each n-tuple $\langle a_{i_1},\ldots,a_{i_n}\rangle \in P_M$, and $\neg\, Px_{i_1}\cdots x_{i_n}$ for each n-tuple $\langle a_{i_1},\ldots,a_{i_n}\rangle \notin P_M$.

On the other hand, we have:

Theorem 4.1.13 (Löwenheim/Skolem Theorem)**.**

1. If S is normally satisfiable in arbitrarily large finite domains, then S is normally satisfiable in some infinite domain.

2. If S is normally satisfiable in some infinite domain, then S is normally satisfiable in all infinite domains of cardinality $\geq$ the cardinality of S.

Proof. For the first part, let $S^* = S \cup \{H_n \mid n = 1,2,\ldots\}$ where H_n is the sentence

$$\exists x_1 \cdots \exists x_n\,\neg\,(Ix_1x_2 \vee Ix_1x_3 \vee \cdots \vee Ix_{n-1}x_n)$$

saying that there exist at least n things. Since S is normally satisfiable in arbitrarily large finite domains, each finite subset of S^* is normally satisfiable. Hence, by Corollary 4.1.9, S^* is normally satisfiable. But any normal L-structure satisfying S^* satisfies S and has an infinite domain.

For the second part, let κ be a cardinal number $\geq$ the cardinality of S. Let $L^* = L \cup \{Q_i \mid i \in X\}$, where each Q_i is a new 1-ary predicate, and X is a set of cardinality κ. Let $S^* = S \cup \{\exists x\, Q_ix \mid i \in X\} \cup \{\neg\,\exists x\,(Q_ix \wedge Q_jx) \mid i,j \in X, i \neq j\} \cup \{\text{identity axioms for } L^*\}$. Thus S^* is a set of L^*-sentences of cardinality κ. Furthermore, any domain in which S^* is satisfiable will contain pairwise distinct elements a_i, $i \in X$, and will therefore have cardinality $\geq \kappa$. By assumption, S is normally satisfiable in some infinite domain. It follows that each finite subset of S^* is satisfiable. Hence, by the Compactness Theorems 2.6.1 and 2.6.2, S^* is satisfiable. Hence, by part 2 of Theorem 2.7.1, S^* is satisfiable in a domain of cardinality κ. Therefore, by Theorems 4.1.5 and 4.1.6, S^* is normally satisfiable in a domain of cardinality $\leq \kappa$, hence $= \kappa$. Let M^* be a normal L^*-structure with U_{M^*} of cardinality κ. Let M be the *reduct* of M^* to L, i.e., M is the L-structure with $U_M = U_{M^*}$ and $P_M = P_{M^*}$ for each predicate P in L. Then M normally satisfies S and U_M is of cardinality κ. $\square$

Exercise 4.1.14. Let L be the following language:

Ox: $x = 1$

$Pxyz$: $x + y = z$

$Qxyz$: $x \times y = z$

Rxy: $x < y$

Sxy: $x + 1 = y$

Ixy: $x = y$ (identity predicate)

For each positive integer n, let M_n be the normal L-structure

$$M_n = (U_n, O_n, P_n, Q_n, R_n, S_n, I_n)$$

where

$U_n = \{1, \ldots, n\}$

$O_n = \{1\}$

$P_n = \{\langle i, j, k\rangle \in (U_n)^3 \mid i + j = k\}$

$Q_n = \{\langle i, j, k\rangle \in (U_n)^3 \mid i \times j = k\}$

$R_n = \{\langle i, j\rangle \in (U_n)^2 \mid i < j\}$

$S_n = \{\langle i, j\rangle \in (U_n)^2 \mid i + 1 = j\}$

$I_n = \{\langle i, j\rangle \in (U_n)^2 \mid i = j\}$

Exhibit an L-sentence Z such that, for all finite normal L-structures M', $M' \models Z$ if and only if M' is isomorphic to M_n for some n.

Solution. As Z we may take the conjunction of the following clauses.

(a) $\forall x \, \forall y \, (Rxy \lor Ryx \lor Ixy)$

(b) $\forall x \, \forall y \, (Rxy \Rightarrow \neg \, Ryx)$

(c) $\forall x \, \forall y \, \forall z \, ((Rxy \land Ryz) \Rightarrow Rxz)$

(d) $\forall x \, \forall z \, (Sxz \Leftrightarrow (Rxz \land \neg \, \exists y \, (Rxy \land Ryz)))$

(e) $\forall u \, (Ou \Leftrightarrow \neg \, \exists x \, Rxu)$

(f) $\forall u\,(Ou \Rightarrow \forall x\,\forall z\,(Puxz \Leftrightarrow Sxz))$

(g) $\forall v\,\forall w\,(Svw \Rightarrow \forall x\,\forall z\,(Pwxz \Leftrightarrow \exists y\,(Syz \wedge Pvxy)))$

(h) $\forall u\,(Ou \Rightarrow \forall x\,\forall z\,(Quxz \Leftrightarrow Ixz))$

(i) $\forall v\,\forall w\,(Svw \Rightarrow \forall x\,\forall z\,(Qwxz \Leftrightarrow \exists y\,(Qvxy \wedge Pxyz)))$

Clauses (a), (b) and (c) say that R is an irreflexive linear ordering of the universe. Clause (d) says that S is the immediate successor relation, with respect to R. Clause (e) says that 1 is the first element of the universe, with respect to R. Clauses (f) and (g) define the addition predicate P, by induction along R, in terms of S. Clauses (h) and (i) define the multiplication predicate Q, by induction along R, in terms of S and P.

Exercise 4.1.15. Let L and M_n be as in Exercise 4.1.14. Show that there exists an infinite normal L-structure $M = M_\infty$ with the following property: for all L-sentences A, if $M_p \models A$ for all sufficiently large primes p, then $M_\infty \models A$. (Hint: Use the Compactness Theorem.)

Solution. Let S be the set of L-sentences A with the following property: there exists $n = n_A$ such that for all primes $p > n_A$, M_p satisfies A. We claim that every finite subset of S is normally satisfiable. To see this, let $S_0 = \{A_1, \ldots, A_k\}$ be a finite subset of S. Put $n = \max(n_{A_1}, \ldots, n_{A_k})$. Let p be any prime $> n$. Then M_p satisfies $A_1, \ldots, A_k$. This proves our claim. By the Compactness Theorem for normal satisfiability (Corollary 4.1.9), it follows that S is normally satisfiable. Let M_∞ be a normal L-structure satisfying S. Among the sentences of S are those asserting that the universe has at least k elements, for each positive integer k. Since M_∞ satisfies these sentences, M_∞ is infinite.

4.2 The Spectrum Problem

Definition 4.2.1. Let A be a sentence of the predicate calculus with identity. The *spectrum of* A is the set of positive integers n such that A is normally satisfiable in a domain of cardinality n. A *spectrum* is a set X of positive integers, such that $X = \text{spectrum}(A)$ for some A.

Remark 4.2.2. The *spectrum problem* is the problem of characterizing the spectra, among all sets of positive integers. This is a famous and apparently difficult open problem.[1] In particular, it is unknown whether the complement of a spectrum is necessarily a spectrum.

Example 4.2.3. We show that the set $\{n \geq 1 \mid n \text{ is even}\}$ is a spectrum.

[1] Jones/Selman [1] show that X is a spectrum if and only if there exists a nondeterministic Turing machine which accepts X in time 2^{ck}, where k is the length of the input. Since the input is a positive integer n, we have $k = [\log_2 n]$, as usual in computational number theory.

Let U be a nonempty set. A binary relation $R \subseteq U^2$ is said to be an *equivalence relation* on U if it is reflexive, symmetric, and transitive, i.e., if the structure (U, R) satisfies $(1) \wedge (2) \wedge (3)$:

$$\begin{array}{ll} (1) & \forall x\, Rxx \\ (2) & \forall x\, \forall y\, (Rxy \Leftrightarrow Ryx) \\ (3) & \forall x\, \forall y\, \forall z\, ((Rxy \wedge Ryz) \Rightarrow Rxz) \end{array}$$

In this situation, the equivalence classes $[a]_R = \{b \in U \mid \langle a, b\rangle \in R\}$, $a \in U$, form a *partition of* U, i.e., a decomposition of the set U into pairwise disjoint, nonempty subsets.

Let A be the following sentence of the predicate calculus with identity:

$$(1)\ \wedge\ (2)\ \wedge\ (3)\ \wedge\ \forall x\, \exists y\, ((\neg\, Ixy) \wedge \forall z\, (Rxz \Leftrightarrow (Ixz \vee Iyz)))$$

Intuitively, A says that R is an equivalence relation with the property that each equivalence class consists of exactly two elements. Obviously, a finite set U admits an equivalence relation with this property if and only if the cardinality of U is even. Thus the spectrum of A is the set of even numbers.

Exercises 4.2.4. Prove the following.

1. If X is a finite or cofinite[2] set of positive integers, then X is a spectrum.

2. The set of even numbers is a spectrum.

3. The set of odd numbers is a spectrum.

4. If r and m are positive integers, $\{n \geq 1 \mid n \equiv r \bmod m\}$ is a spectrum.

5. If X and Y are spectra, $X \cup Y$ and $X \cap Y$ are spectra.

Solution.

1. Let E_n be sentence in the language with only the identity predicate I, saying that the universe consists of exactly n elements (Exercise 4.1.10). If $X = \{n_1, \ldots, n_k\}$, then X is the spectrum of $E_{n_1} \vee \cdots \vee E_{n_k}$, and the complement of X is spectrum of $\neg\, (E_{n_1} \vee \cdots \vee E_{n_k})$.

2. The even numbers are the spectrum of a sentence which says: R is an equivalence relation on the universe, such that each equivalence class consists of exactly two elements. For more details, see Example 4.2.3.

3. The odd numbers are the spectrum of a sentence which says: R is an equivalence relation on the universe, such that each equivalence class consists of exactly two elements, except for one equivalence class, which consists of exactly one element.

4. We may assume that $0 \leq r < m$. If $r = 0$, the set

[2] A set of positive integers is said to be *cofinite* if its complement is finite.

$$\{n \geq 1 : n \equiv 0 \bmod m\} = \{n \geq 1 : m \text{ divides } n \text{ with no remainder}\}$$

is the spectrum of a sentence which says: R is an equivalence relation on the universe, such that each equivalence class consists of exactly m elements. If $r > 0$, the set

$$\{n \geq 1 : n \equiv r \bmod m\} = \{n \geq 1 : m \text{ divides } n \text{ with remainder } r\}$$

is the spectrum of a sentence which says: R is an equivalence relation on the universe, such that each equivalence class consists of exactly m elements, except for one equivalence class, which consists of exactly r elements.

5. Assume that X is the spectrum of A and Y is the spectrum of B. Then $X \cup Y$ is the spectrum of $A \lor B$. Also, $X \cap Y$ is the spectrum of $A \land B$, provided A and B have no predicates in common except the identity predicate. To arrange for this, replace B by an analogous sentence in a different language.

Exercise 4.2.5. Prove that, for any sentence A of the predicate calculus with identity, at least one of spectrum(A) and spectrum($\neg A$) is cofinite. (Hint: Use part 1 of Theorem 4.1.13.)

Example 4.2.6. We show that the set of composite numbers[3] is a spectrum.

Let L be a language consisting of two binary predicates, R and S, as well as the identity predicate, I. Let A be an L-sentence saying that R and S are equivalence relations, each with more than one equivalence class, and

$$\forall x\, \forall y\, (\exists \text{ exactly one } z)(Rxz \land Syz).$$

Thus, for any normal L-structure $M = (U_M, R_M, S_M, I_M)$ satisfying A, we have that R_M and S_M partition U_M into "rows" and "columns", respectively, in such a way that the intersection of any "row" with any "column" consists of exactly one element of U_M. Thus, if U_M is finite, the elements of U_M are arranged in an $m \times n$ "matrix", where $m, n \geq 2$. Therefore, the number of elements in U_M is mn, a composite number. Conversely, for any $m, n \geq 2$, there is an L-structure M as above, which satisfies A. Thus spectrum(A) is the set of composite numbers.

Exercise 4.2.7. Use the result of Exercise 4.1.14 to prove the following:

1. The set of prime numbers and its complement are spectra.

2. The set of squares $\{1, 4, 9, \ldots\}$ and its complement are spectra.

3. The set of powers of 2, $\{2^n \mid n = 1, 2, 3, \ldots\}$, and its complement, are spectra.

[3] A *composite number* is an integer greater than 1 which is not prime.

4. The set of prime powers $\{p^n \mid p \text{ prime}, n = 1, 2, \ldots\}$ and its complement are spectra.

Solution. Let Z be as in Exercise 4.1.14 above. For each of the given sets X, we exhibit a sentence A with the following properties: X is the spectrum of $Z \wedge A$, and the complement of X is the spectrum of $Z \wedge \neg A$.

1. $\exists z\, ((\neg \exists w\, Rzw) \wedge (\neg \exists x\, \exists y\, (Rxz \wedge Ryz \wedge Qxyz)) \wedge (\neg Oz))$.
2. $\exists z\, ((\neg \exists w\, Rzw) \wedge \exists x\, Qxxz)$.
3. $\exists z\, \exists v\, ((\neg \exists w\, Rzw) \wedge (\exists u\, (Ou \wedge Suv))$
   $\wedge \forall x\, ((\neg Ox \wedge \exists y\, Qxyz) \Rightarrow \exists w\, Qvwx))$.
4. $\exists z\, \exists v\, ((\neg \exists w\, Rzw) \wedge (\neg \exists x\, \exists y\, (Rxv \wedge Ryv \wedge Qxyv)) \wedge (\neg Ov)$
   $\wedge \forall x\, ((\neg Ox \wedge \exists y\, Qxyz) \Rightarrow \exists w\, Qvwx))$.

Exercise 4.2.8.

1. The Fibonacci numbers are defined recursively by $F_1 = 1$, $F_2 = 2$, $F_n = F_{n-1} + F_{n-2}$ for $n \geq 3$. Show that the set of Fibonacci numbers

$$\{F_n \mid n = 1, 2, \ldots\} = \{1, 2, 3, 5, 8, 13, 21, 34, 55, \ldots\}$$

   and its complement are spectra.

2. Show that $\{x^y \mid x, y \geq 2\}$ and its complement are spectra.

Exercise 4.2.9. Let X be a subset of $\{1, 2, 3, \ldots\}$. Prove that if X is a spectrum then $\{n^2 \mid n \in X\}$ is a spectrum.

4.3 Predicate Calculus With Operations

In this section we extend the syntax and semantics of the predicate calculus, to encompass operations. As examples of operations, we may cite the familiar mathematical operations of addition $(+)$ and multiplication $(\times)$. Such operations are considered binary, because they take two arguments. More generally, we consider n-ary operations.

Definition 4.3.1 (languages)**.** A *language* is a set of predicates $P, Q, R, \ldots$ and *operations* $f, g, h, \ldots$. Each predicate and each operation is designated as *n-ary* for some nonnegative[4] integer n.

Definition 4.3.2 (terms, formulas, sentences)**.** Let L be a language, and let U be a set. The set of *L-U-terms* is generated as follows.

1. Each variable is an L-U-term.

[4] A 0-ary operation is known as a *constant*. Syntactically, constants behave as parameters.

2. Each element of U is an L-U-term.

3. If f is an n-ary operation of L, and if $t_1, \ldots, t_n$ are L-U-terms, then $ft_1 \cdots t_n$ is an L-U-term.

An L-U-term is said to be *variable-free* if no variables occur in it. An *atomic L-U-formula* is an expression of the form

$$Pt_1 \cdots t_n$$

where P is an n-ary predicate of L, and $t_1, \ldots, t_n$ are L-U-terms. The set of *L-U-formulas* is generated as in clauses 2, 3, 4 and 5 of Definition 2.1.3. The notions of substitution, free variables, and L-U-sentences are defined as in Section 2.1. Note that $Pt_1 \cdots t_n$ is a sentence if and only if it is variable-free.

Definition 4.3.3 (structures)**.** An *L-structure* M consists of a nonempty set U_M, an n-ary relation $P_M \subseteq (U_M)^n$ for each n-ary predicate P of L, and an n-ary function $f_M : (U_M)^n \to U_M$ for each n-ary operation f of L.

Definition 4.3.4 (isomorphism)**.** Two L-structures M and M' are said to be *isomorphic* if there exists an *isomorphism* of M onto M', i.e., a one-to-one correspondence $\phi : U_M \cong U_{M'}$ such that:

1. for all n-ary predicates P of L and all n-tuples $\langle a_1, \ldots, a_n \rangle \in (U_M)^n$, $\langle a_1, \ldots, a_n \rangle \in P_M$ if and only if $\langle \phi(a_1), \ldots, \phi(a_n) \rangle \in P_{M'}$.

2. for all n-ary operations f of L and all n-tuples $\langle a_1, \ldots, a_n \rangle \in (U_M)^n$, $\phi(f_M(a_1, \ldots, a_n) = f_{M'}(\phi(a_1), \ldots, \phi(a_n))$.

Lemma 4.3.5 (valuations)**.** Let M be an L-structure.

1. There is a unique valuation

$$v_M : \{t \mid t \text{ is a variable-free } L\text{-}U_M\text{-term}\} \to U_M$$

defined as follows:

(a) $v_M(a) = a$ for all $a \in U_M$.
(b) $v_M(ft_1 \cdots t_n) = f_M(v_M(t_1), \ldots, v_M(t_n))$ for all n-ary operations f of L and all variable-free L-U_M-terms $t_1, \ldots, t_n$.

2. There is a unique valuation

$$v_M : \{A \mid A \text{ is an } L\text{-}U_M\text{-sentence}\} \to \{\mathrm{T}, \mathrm{F}\}$$

defined as follows. For atomic L-U-sentences, we have

$$v_M(Pt_1 \cdots t_n) = \begin{cases} \mathrm{T} & \text{if } \langle v_M(t_1), \ldots, v_M(t_n) \rangle \in P_M, \\ \mathrm{F} & \text{if } \langle v_M(t_1), \ldots, v_M(t_n) \rangle \notin P_M. \end{cases}$$

For non-atomic L-U_M-sentences, $v_M(A)$ is defined as in clauses 2 through 8 of Lemma 2.2.4.

Proof. The proof is as for Lemma 2.2.4. □

Definition 4.3.6 (tableau method)**.** The signed and unsigned tableau methods carry over to predicate calculus with operations. We modify the tableau rules as follows.

Signed:

$$
\begin{array}{ccc}
\vdots & \qquad\qquad & \vdots \\
\mathrm{T}\,\forall x\, A & & \mathrm{F}\,\exists x\, A \\
\vdots & & \vdots \\
| & & | \\
\mathrm{T}\, A[x/t] & & \mathrm{F}\, A[x/t]
\end{array}
$$

where t is a variable-free term

$$
\begin{array}{ccc}
\vdots & \qquad\qquad & \vdots \\
\mathrm{T}\,\exists x\, A & & \mathrm{F}\,\forall x\, A \\
\vdots & & \vdots \\
| & & | \\
\mathrm{T}\, A[x/a] & & \mathrm{F}\, A[x/a]
\end{array}
$$

where a is a new parameter

Unsigned:

$$
\begin{array}{ccc}
\vdots & \qquad\qquad & \vdots \\
\forall x\, A & & \neg\,\exists x\, A \\
\vdots & & \vdots \\
| & & | \\
A[x/t] & & \neg\, A[x/t]
\end{array}
$$

where t is a variable-free term

$$
\begin{array}{ccc}
\vdots & \qquad\qquad & \vdots \\
\exists x\, A & & \neg\,\forall x\, A \\
\vdots & & \vdots \\
| & & | \\
A[x/a] & & \neg\, A[x/a]
\end{array}
$$

where a is a new parameter

Remark 4.3.7 (soundness and completeness)**.** With the tableau rules as above, the Soundness Theorem 2.3.13 carries over unchanged to the context of predicate calculus with operations. The results of Section 2.4 on logical equivalence also carry over. The notion of U-repleteness (Definition 2.5.2) is modified to say that, for example, if S contains $\forall x\, A$ then S contains $A[x/t]$ for all variable-free L-U-terms t. The conclusion of Hintikka's Lemma 2.5.3 is modified to say that S is satisfiable in the domain of variable-free L-U-terms. The conclusion of the Completeness Theorem 2.5.5 is modified to say that $X_1, \ldots, X_k$ is satisfiable in the domain of variable-free L-V-terms. The Compactness Theorems 2.6.1 and 2.6.2 carry over unchanged.

Remark 4.3.8 (satisfiability in a domain)**.** The notion of satisfiability in a domain carries over unchanged to the context of predicate calculus with operations. Theorems 2.2.6 and 2.2.11 on isomorphism, and Theorem 2.7.1 on satisfiability in infinite domains, also carry over. Theorem 2.7.3 carries over in an appropriately modified form. See Theorem 4.3.9 and Exercise 4.3.10 below.

Theorem 4.3.9. Let M and M' be L-structures. Assume that $\phi : U_M \to U_{M'}$ is an onto mapping such that conditions 1 and 2 of Definition 4.3.4 hold. Then as in Theorem 2.2.6 we have $v_M(A) = v_{M'}(A')$ for all L-U_M-sentences A, where $A' = A[a_1/\phi(a_1), \ldots, a_k/\phi(a_k)]$. In particular, M and M' satisfy the same L-sentences.

Proof. The proof is similar to that of Theorem 2.7.3. □

Exercise 4.3.10. Use Theorem 4.3.9 to show that Corollary 2.7.4 carries over to the context of predicate calculus with operations.

Remark 4.3.11 (companions and proof systems)**.** In our notion of companion (Definition 3.2.3), clauses (1) and (4) are modified as follows:

$$(1) \qquad (\forall x\, B) \Rightarrow B[x/t]$$
$$(4) \qquad B[x/t] \Rightarrow (\exists x\, B)$$

where t is any variable-free term. In our Hilbert-style proof system LH, the instantiation rules are modified as follows:

(a) $(\forall x\, B) \Rightarrow B[x/t]$ (universal instantiation)

(b) $B[x/t] \Rightarrow (\exists x\, B)$ (existential instantiation)

where t is any variable-free term. Also, our Gentzen-style proof system LG is modified in accordance with the modified tableau rules. With these changes, the soundness and completeness of LG and LH carry over.

Exercise 4.3.12 (the Interpolation Theorem)**.** Strengthen the Interpolation Theorem 3.5.1 to say that each operation, predicate and parameter occurring in I occurs in both A and B. (Hint: The version with operations can be deduced from the version without operations.)

Exercise 4.3.13. Skolemization.

4.4 Predicate Calculus with Identity and Operations

Remark 4.4.1 (predicate calculus with identity and operations)**.** We augment the identity axioms (Definition 4.1.2) as follows:

5. For each n-ary operation f of L, we have an axiom

$\forall x_1 \cdots \forall x_n \, \forall y_1 \cdots \forall y_n \, ((Ix_1y_1 \land \cdots \land Ix_ny_n) \Rightarrow Ifx_1 \cdots x_n fy_1 \cdots y_n)$.

The notions of normal structure and normal satisfiability are defined as before. The results of Section 4.1 on the predicate calculus with identity carry over unchanged to the predicate calculus with identity and operations. See also Exercise 4.4.2 below.

Exercise 4.4.2 (elimination of operations)**.** Let L be a language with identity and operations. Let L^o be the language with identity and without operations, obtained by replacing each n-ary operation f belonging to L by a new $(n+1)$-ary predicate P_f belonging to L^o. Each normal L-structure M gives rise to a normal L^o-structure M^o where

$$(P_f)_{M^o} = \{\langle a_1, \ldots, a_n, b\rangle \in (U_M)^{n+1} \mid f_M(a_1, \ldots, a_n) = b\}.$$

For each n-ary operation f of L, let G_f be the L^o-sentence

$$\forall x_1 \cdots \forall x_n \, \exists y \, \forall z \, (Iyz \Leftrightarrow P_f x_1 \cdots x_n z).$$

1. Show that to each L-sentence A we may associate an L^o-sentence A^o such that, for all L-structures M, $M \models A$ if and only if $M^o \models A^o$.

2. Show that a normal L^o-structure satisfies the sentences G_f, f in L, if and only if it is of the form M^o for some L-structure M.

Exercise 4.4.3. Show that the spectrum problem for predicate calculus with identity and operations is equivalent to the spectrum problem for predicate calculus with identity and without operations, as previously discussed in Section 4.2. In other words, given a sentence A involving some operations, construct a sentence A^{oo} involving no operations, such that spectrum$(A) =$ spectrum(A^{oo}). (Hint: Use the result of Exercise 4.4.2. Note that A^{oo} will not be the same as the A^o of Exercise 4.4.2.)

Remark 4.4.4 (predicate calculus with equality)**.** The predicate calculus with identity and operations is well suited for the study of algebraic structures such as number systems, groups, rings, etc. In such a context, one often writes $t_1 = t_2$ instead of It_1t_2, and one refers to *predicate calculus with equality* rather than predicate calculus with identity. In this notation, the *equality axioms* (i.e., the identity axioms) read as follows:

$\forall x \, (x = x)$,

$\forall x\, \forall y\, (x = y \Leftrightarrow y = x)$,

$\forall x\, \forall y\, \forall z\, ((x = y \wedge y = z) \Rightarrow x = z)$,

$\forall x_1\, \forall y_1 \cdots \forall x_n\, \forall y_n\, ((x_1 = y_1 \wedge \cdots \wedge x_n = y_n) \Rightarrow (Px_1 \cdots x_n \Leftrightarrow Py_1 \cdots y_n))$, for each n-ary predicate P,

$\forall x_1\, \forall y_1 \cdots \forall x_n\, \forall y_n\, ((x_1 = y_1 \wedge \cdots \wedge x_n = y_n) \Rightarrow fx_1 \cdots x_n = fy_1 \cdots y_n)$, for each n-ary operation f .

One also uses customary algebraic notation, e.g., $t_1 + t_2$ instead of $+t_1t_2$, $t_1 \times t_2$ or t_1t_2 instead of $\times t_1t_2$, etc. To avoid ambiguity, parentheses are used.

Examples 4.4.5 (groups and rings)**.** Using predicate calculus with identity and operations, a group may be viewed as a normal L-structure

$$G = (U_G, f_G, i_G, e_G, I_G).$$

Here U_G is the underlying set of the group, and L is the language $\{f, i, e, I\}$, where f is the group composition law (a binary operation), i is group inversion (a unary operation), e is the group identity element (a 0-ary operation, i.e., a constant), and I is the identity predicate (a binary predicate). We could refer to L as *the language of groups*. It is customary to write G instead of U_G, $t_1 \cdot t_2$ or t_1t_2 instead of ft_1t_2, t^{-1} instead of it, 1 instead of e, and $t_1 = t_2$ instead of It_1t_2. Thus

$$G = (G, \cdot_G, {}^{-1}{}_G, 1_G, =_G)$$

and G is required to satisfy the *group axioms*, consisting of the identity axioms for L, plus $\forall x\, \forall y\, \forall z\, ((xy)z = x(yz))$, $\forall x\, (x^{-1}x = xx^{-1} = 1)$, $\forall x\, (1x = x1 = x)$.

Similarly, a ring may be viewed as a normal structure

$$R = (R, +_R, \cdot_R, -_R, 0_R, 1_R, =_R)$$

where $+$ and $\cdot$ are binary operations, $-$ is a unary operation, 0 and 1 are constants, and $=$ is the equality predicate. We could refer to the language $\{+, \cdot, -, 0, 1, =\}$ as *the language of rings*. R is required to satisfy the *ring axioms*, consisting of the identity axioms plus $\forall x\, \forall y\, \forall y\, (x + (y + z) = (x + y) + z)$, $\forall x\, \forall y\, (x + y = y + x)$, $\forall x\, (x + 0 = x)$, $\forall x\, (x + (-x) = 0)$, $\forall x\, \forall y\, \forall z\, (x \cdot (y \cdot z) = (x \cdot y) \cdot z)$, $\forall x\, (x \cdot 1 = 1 \cdot x = x)$, $\forall x\, \forall y\, \forall z\, (x \cdot (y + z) = (x \cdot y) + (x \cdot z))$, $\forall x\, \forall y\, \forall z\, ((x + y) \cdot z = (x \cdot z) + (y \cdot z))$, $0 \neq 1$.

Exercise 4.4.6. Let G be a group. For $a \in G$ write $a^n = a \cdot \cdots \cdot a$ (n times). Thus $a^1 = a$ and $a^{n+1} = a^n \cdot a$. We say that G is a *torsion group* if for all $a \in G$ there exists a positive integer n such that $a^n = 1$. We say that G is *torsion-free* if for all $a \in G$, if $a \neq 1$ then $a^n \neq 1$ for all positive integers n.

1. Show that the class of torsion-free groups can be characterized by a set of sentences. I.e., there is a set of sentences S such that, for all groups G, G is torsion-free if and only if $G \models S$.

   Solution. Let $S = \{A_n : n \geq 2\}$, where A_n is the sentence

$$\forall x\,(x \neq 1 \Rightarrow x^n \neq 1).$$

Clearly the groups satisfying S are exactly the torsion-free groups.

2. Show that the class of torsion-free groups cannot be characterized by a finite set of sentences.

   Solution. Suppose S' were a finite of sentences such that the groups satisfying S' are exactly the torsion-free groups. In particular, each sentence in S' is a logical consequence of the group axioms plus $S = \{A_n : n \geq 2\}$ as above. By the Compactness Theorem, each sentence in S' is a logical consequence of the group axioms plus $\{A_2, \ldots, A_n\}$ for sufficiently large n. Since S' is finite, there is a fixed n such that all of the sentences in S' are logical consequences of the group axioms plus $\{A_2, \ldots, A_n\}$. Now let G be a torsion group satisfying $\{A_2, \ldots, A_n\}$. (For example, we may take G to be the additive group of integers modulo p, where p is a prime number greater than n.) Then G satisfies S' yet is not torsion-free, contradicting our assumption on S'.

3. Show that the class of torsion groups cannot be characterized by a set of sentences. I.e., there is no set of sentences S with the property that, for all groups G, G is a torsion group if and only if $G \models S$.

Exercise 4.4.7. Let L be the language of groups. Let S be the set of L-sentences which are true in all finite groups. Define a *pseudo-finite group* to be a group which satisfies S. Note that every finite group is pseudo-finite.

Does there exist an infinite, pseudo-finite group? Prove your answer.

4.5 Many-Sorted Predicate Calculus

Definition 4.5.1 (many-sorted languages)**.** A *many-sorted language* consists of

1. a set of *sorts* $\sigma, \tau, \ldots$,
2. a set of predicates $P, Q, \ldots$, each designated as *n-ary of type* $(\sigma_1, \ldots, \sigma_n)$ for some nonnegative integer n and sorts $\sigma_1, \ldots, \sigma_n$,
3. a set of operations $f, g, \ldots$, each designated as *n-ary of type* $(\sigma_1, \ldots, \sigma_n, \tau)$ for some nonnegative integer n and sorts $\sigma_1, \ldots, \sigma_n, \tau$.

Definition 4.5.2 (terms, formulas, sentences)**.** Let L be a many-sorted language. For each sort σ, we assume a fixed, countably infinite set of *variables of sort* σ, denoted $x^\sigma, y^\sigma, z^\sigma, \ldots$. Let $U = (U^\sigma, U^\tau, \ldots)$ consist of a set U^σ for each sort σ of L. The *L-U-terms* are generated as follows.

1. Each variable of sort σ is a term of sort σ.

2. Each element of U^σ is a term of sort σ.

3. If f is an n-ary operaton of type $(\sigma_1, \ldots, \sigma_n, \tau)$, and if $t_1, \ldots, t_n$ are terms of sort $\sigma_1, \ldots, \sigma_n$ respectively, then $ft_1 \ldots t_n$ is a term of sort τ.

An *atomic L-U-formula* is an expression of the form $Pt_1 \ldots t_n$, where P is an n-ary predicate of type $(\sigma_1, \ldots, \sigma_n)$, and $t_1, \ldots, t_n$ are terms of sort $\sigma_1, \ldots, \sigma_n$ respectively. The *L-U-formulas* are generated as in Definition 2.1.3, with clause 5 modified as follows:

5′. If x^σ is a variable of sort σ, and if A is an L-U-formula, then $\forall x^\sigma\, A$ and $\exists x^\sigma\, A$ are L-U-formulas.

Our notions of substitution, free and bound variables, sentences, etc., are extended in the obvious way to the many-sorted context. Naturally, the substitution $A[x^\sigma/t]$ makes sense only when t is a term of sort σ. An *L-formula* is an L-U-formula where $U_\sigma = \emptyset$ for each sort σ.

Definition 4.5.3 (many-sorted structures)**.** An *L-structure M* consists of

1. a nonempty set U_M^σ for each sort σ of L,

2. an n-ary relation $P_M \subseteq U_M^{\sigma_1} \times \cdots \times U_M^{\sigma_n}$ for each n-ary predicate P of type $(\sigma_1, \ldots, \sigma_n)$ belonging to L,

3. an n-ary function $f_M : U_M^{\sigma_1} \times \cdots \times U_M^{\sigma_n} \to U_M^\tau$ for each n-ary operation f of type $(\sigma_1, \ldots, \sigma_n, \tau)$ belonging to L.

Notions such as isomorphism, valuation, truth, satisfiability, and results such as Theorem 2.2.6 on isomorphism, and Theorem 4.3.9 on onto mappings, carry over to the many-sorted context in the obvious way.

Definition 4.5.4 (many-sorted domains)**.** We define a *domain* or *universe* for L to be an indexed family of nonempty sets $U = (U^\sigma, U^\tau, \ldots)$, where $\sigma, \tau, \ldots$ are the sorts of L. In this way, the notion of satisfiability in a domain generalizes to the many-sorted context.

Remark 4.5.5 (tableau method, proof systems)**.** For each sort σ of L, fix a countably infinite set $V^\sigma = \{a^\sigma, b^\sigma, \ldots\}$, the set of *parameters of sort* σ. Then the tableau method carries over in the obvious way, generalizing Remark 4.3.7. In the Completeness Theorem for the tableau method, we obtain satisfiability in the domain $U = (U^\sigma, U^\tau, \ldots)$, where U^σ is the set of variable-free L-V-terms of sort σ, with $V = (V^\sigma, V^\tau, \ldots)$. The soundness and completeness of our proof systems LH and LG and the Interpolation Theorem also carry over, just as in Section 4.3.

Remark 4.5.6 (identity predicates)**.** For each sort σ of L, L may or may not contain a binary predicate I^σ of type (σ, σ) designated as the *identity predicate* for σ. As *identity axioms* we may take the universal closures of all L-formulas of the form

$$\forall x^\sigma\, \forall y^\sigma\, (I^\sigma xy \Rightarrow (A \Leftrightarrow A[x/y]))$$

where A is atomic. An L-structure M is said to be *normal* if $I^\sigma_M = \{\langle a, a\rangle \mid a \in U^\sigma_M\}$ for all σ such that I^σ belongs to L. The results of Section 4.1 concerning normal satisfiability carry over to the many-sorted context.

Definition 4.5.7 (languages with identity)**.** A *many-sorted language with identity* is a many-sorted language which contains an identity predicate for each sort.

Remark 4.5.8 (many-sorted spectrum problem)**.** Let L be a many-sorted language with identity. If A is an L-sentence and $\sigma_1, \ldots, \sigma_k$ are the sorts occurring in A, the *spectrum* of A is the set of k-tuples of positive integers $(n_1, \ldots, n_k)$ such that there exists a normal L-structure M with $U^{\sigma_i}_M$ of cardinality n_i, for $i = 1, \ldots, k$. In this way, the spectrum problem carries over to many-sorted predicate calculus. So far as I know, the problem of characterizing many-sorted spectra has not been investigated thoroughly.

Remark 4.5.9 (many-sorted Löwenheim/Skolem theorems)**.** It is natural to try to generalize the Löwenheim/Skolem Theorem 4.1.13 to many-sorted predicate calculus. This is straightforward provided we consider only normal structures M where all of the domains $U^\sigma_M, U^\tau_M, \ldots$ are of the same infinite cardinality. However, if we require $U^\sigma_M, U^\tau_M, \ldots$ to be of specified distinct cardinalities, then this leads to difficult issues. Even for two sorts, the topic of so-called *two-cardinal theorems* turns out to be rather delicate and complicated. See for example the model theory textbook of Marker [2].

Remark 4.5.10. Our reasons for including many-sorted predicate calculus in this course are as follows:

1. it is more useful ....

   FIXME

Remark 4.5.11 (one-sorted languages)**.** A language or structure is said to be *one-sorted* if it has only one sort. This term is used for contrast with the many-sorted generalization which we are considering in this section. Generally speaking, one-sorted logic tends to be a little simpler than many-sorted logic.

Chapter 5

Theories, Models, Definability

5.1 Theories and Models

Definition 5.1.1. A *theory* T consists of a language L, called the *language of* T, together with a set of L-sentences called the *axioms of* T. Thus $T = (L, S)$, where L is the language of T, and S is the set of axioms of T.

Definition 5.1.2. Let $T = (L, S)$ be a theory.

1. A *model of* T is an L-structure M such that M satisfies S. If L contains identity predicates, then M is required to be normal with respect to these predicates.

2. A *theorem* of T is an L-sentence A such that A is true in all models of T, i.e., A is a logical consequence of the axioms of T. Equivalently, A is derivable in $LH(S \cup \{\text{identity axioms for } L\})$.

   If A is a theorem of T, we denote this by $T \vdash A$.

3. T is *finitely axiomatized* if S is finite.

4. Two theories are *equivalent* if they have the same language and the same theorems. I.e., they have the same language and the same models.

5. T is *finitely axiomatizable* if it is equivalent to a finitely axiomatized theory.

Definition 5.1.3 (consistency, categoricity, completeness)**.** Let $T = (L, S)$ be a theory.

1. T is *consistent* if there exists at least one model of T. Equivalently, T is consistent if and only if there is no L-sentence A such that both $T \vdash A$ and $T \vdash \neg A$. Equivalently, T is consistent if and only if there exists an L-sentence A such that $T \nvdash A$.

2. T is *categorical* if T is consistent and all models of T are isomorphic.

3. T is *complete* if T is consistent and all models of T are elementarily equivalent. Equivalently, T is complete if and only if for all L-sentences A either $T \vdash A$ or $T \vdash \neg A$ but not both.

Remark 5.1.4. Our formal notion of *theory*, as defined above, is intended as a precise explication of the informal notion of "deductive scientific theory". The language of T is the vocabulary of our theory. The theorems of T are the assertions of our theory. The axioms of T are the basic assertions, from which all others are deduced. Consistency of T means that our theory is free of internal contradictions. Categoricity of T means that our theory is fully successful in that it fully captures the structure of the underlying reality described by the theory. Completeness of T means that our theory is sufficiently successful to decide the truth values of all statements expressible in the language of the theory.

Remark 5.1.5 (mathematical theories, foundational theories)**.** Later in this chapter we shall present several interesting examples of theories. Loosely speaking, the examples are of two kinds.

The first kind consists of *mathematical theories*. By a mathematical theory we mean a theory which is introduced in order to describe a certain class of mathematical structures. See Section 5.2. Since the class is diverse, the theory is typically not intended to be complete. Nevertheless we shall see that, remarkably, several of these mathematical theories turn out to be complete.

The second kind consists of *foundational theories*, i.e., theories which are introduced in order to serve as a general axiomatic foundation for all of mathematics, or at least a large part of it. See Sections 5.5 and 5.6. Each such theory is intended to fully describe a specific, foundationally important model, known as the *intended model* of the theory. Although such theories are intended to be complete, we shall see in Chapter 6 that, regrettably, most of these theories turn out to be incomplete.

One way to show that a theory is complete is to show that it is categorical.

Theorem 5.1.6. If T is categorical, then (1) T is complete, and (2) the language of T is a language with identity.

Proof. Assume that T is categorical. Then any two models of T are isomorphic. Hence by Theorem 2.2.6 (see also Definition 4.5.3), any two models of T are elementarily equivalent. Hence T is complete. Also, by Theorem 2.7.3 (see also Theorem 4.3.9 and Definition 4.5.3), T contains an identity predicate for each sort in the language of T. $\square$

On the other hand, we have:

Exercise 5.1.7. Let T be a complete theory in a language with identity. Let M be a model of T.

1. In the one-sorted case, show that T is categorical if and only if the universe U_M is finite.

2. In the many-sorted case with sorts $\sigma, \tau, \ldots$, show that T is categorical if and only if each of the universes $U_M^\sigma, U_M^\tau, \ldots$ is finite.

Solution. If one of the universes U_M^σ is infinite, we can use the Löwenheim/Skolem Theorem 4.1.13 to blow it up to an arbitrarily large uncountable cardinality.

Remark 5.1.8. Theorem 5.1.6 and Exercise 5.1.7 show that we cannot use categoricity as a test for completeness, except in very special circumstances. A similar but more useful test is provided by the following theorem.

Definition 5.1.9. Let κ be an infinite cardinal number. A one-sorted theory T is said to be *κ-categorical* if all models of T of cardinality κ are isomorphic.

Theorem 5.1.10 (Vaught's Test)**.** Let T be a one-sorted theory. Assume that (a) T is consistent, (b) all models of T are infinite, and (c) there exists an infinite cardinal $\kappa \geq$ the cardinality of the language of T such that T is κ-categorical. Then T is complete.

Proof. Suppose T is not complete. Since T is consistent, there exist M_1 and M_2 which are models of T and not elementarily equivalent. By assumption (b), M_1 and M_2 are infinite. Let κ be an infinite cardinal $\geq$ the cardinality of the language of T. By the Löwenheim/Skolem Theorem 4.1.13, there exist models M_1', M_2' of cardinality κ elementarily equivalent to M_1, M_2 respectively. Clearly M_1', M_2' are not elementarily equivalent. Hence, by Theorem 2.2.6, M_1' and M_2' are not isomorphic. This contradicts assumption (c). $\square$

Exercise 5.1.11. Generalize Vaught's Test to many-sorted theories.

5.2 Mathematical Theories

In this section we give several examples of theories suggested by abstract algebra and other specific mathematical topics. We point out that several of these mathematical theories are complete.

Example 5.2.1 (groups)**.** The *language of groups* consists of a binary operation $\cdot$ (multiplication), a unary operation $^{-1}$ (inverse), a constant 1 (the identity element), and a binary predicate $=$ (equality). The *theory of groups* consists of the equality axioms plus

$\forall x \, \forall y \, \forall z \, (x \cdot (y \cdot z) = (x \cdot y) \cdot z)$ (associativity),

$\forall x \, (x \cdot x^{-1} = x^{-1} \cdot x = 1)$ (inverses),

$\forall x \, (x \cdot 1 = 1 \cdot x = x)$ (identity).

A *group* is a model of the theory of groups. A group is said to be *Abelian* if it satisfies the additional axiom

$\forall x \, \forall y \, (x \cdot y = y \cdot x)$ (commutativity).

A *torsion group* is a group G such that for all $a \in G$ there exists a positive integer n such that $a^n = 1$. A group G is *torsion-free* if for all $a \in G$, if $a \neq 1$ then $a^n \neq 1$ for all positive integers n. Note that G is torsion-free if and only if it satisfies the axioms $\forall x\,(x^n = 1 \Rightarrow x = 1)$ for $n = 2, 3, \ldots$. A group is said to be *divisible* if it satisfies the axioms $\forall x\,\exists y\,(y^n = x)$, for $n = 2, 3, \ldots$.

Exercise 5.2.2. Show that the theory of torsion-free Abelian groups is not finitely axiomatizable. Deduce that the theory of torsion-free groups is not finitely axiomatizable.

Solution. Suppose that the theory of torsion-free Abelian groups were finitely axiomatizable. By Compactness, the axioms would be logical consequences of the Abelian group axioms plus finitely many axioms of the form $\forall x\,(x^n = 1 \Rightarrow x = 1)$, say $n = 1, \ldots, k$. Let p be a prime number greater than k. Then the additive group of integers modulo p satisfies these axioms but is not torsion-free. This is a contradiction.

If the theory of torsion-free groups were finitely axiomatizable, then the theory of torsion-free Abelian groups would also be finitely axiomatizable, by adjoining the single axiom $\forall x\,\forall y\,(x \cdot y = y \cdot x)$.

Exercise 5.2.3. Show that there exist Abelian groups G_1 and G_2 such that G_1 is a torsion group, G_1 is elementarily equivalent to G_2, yet G_2 is not a torsion group. Deduce that there is no theory in the language of groups whose models are precisely the Abelian torsion groups. Hence, there is no theory in the language of groups whose models are precisely the torsion groups.

Solution. Let G_1 be a torsion group with elements of arbitrarily large finite order. (For example, we could take G_1 to be the additive group of rational numbers modulo 1.) Let L be the language of groups, and let S be the set of all L-sentences true in G_1. Let $L^* = L \cup \{c\}$ where c is a new constant, and let $S^* = S \cup \{c^n \neq 1 \mid n = 1, 2, \ldots\}$. By choosing $c \in G_1$ appropriately, we see that any finite subset of S^* is normally satisfiable. By the Compactness Theorem for normal satisfiability (Corollary 4.1.9), it follows that S^* is normally satisfiable, so let (G_2, c) be a model of S^*. Then G_2 is an Abelian group which is elementarily equivalent to G_1 yet contains an element c of infinite order, hence is not a torsion group.

If T were an L-theory whose models are just the Abelian torsion groups, then G_1 would be a model of T but G_2 would not, contradicting the fact that G_1 and G_2 are elementarily equivalent.

If T were an L-theory whose models are just the torsion groups, then $T \cup \{\forall x\,\forall y\,(x \cdot y = y \cdot x)\}$ would be an L-theory whose models are just the Abelian torsion groups.

Remark 5.2.4. Let DAG_0 be the theory of infinite torsion-free divisible Abelian groups. It can be shown that DAG_0 is κ-categorical for all uncountable cardinals κ. (This is because such groups may be viewed as vector spaces over the rational field, $\mathbb{Q}$.) It follows by Vaught's Test that DAG_0 is complete.

Example 5.2.5 (linear orderings)**.** The *language of linear orderings* consists of a binary predicate $<$ plus the equality predicate $=$. The axioms for linear orderings are $\forall x\, \forall y\, \forall z\, ((x < y \wedge y < z) \Rightarrow x < z)$, and $\forall x\, (\neg\, x < x)$, and $\forall x\, \forall y\, (x < y \vee x = y \vee x > y)$. A *linear ordering* is a model of these axioms.

A linear ordering is said to be *nontrivial* if it satisfies $\exists x\, \exists y\, (x < y)$. It is said to be *dense* if it is nontrivial and satisfies $\forall x\, \forall y\, (x < y \Rightarrow \exists z\, (x < z < y))$. It is said to be *without end points* if it satisfies $\forall x\, \exists y\, (y < x)$ and $\forall x\, \exists y\, (y > x)$. It is said to *with end points* if it satisfies $\exists x\, \neg\, \exists y\, (y < x)$ and $\exists x\, \neg\, \exists y\, (y > x)$. An example of a dense linear ordering without end points is $(\mathbb{Q}, <)$, where $\mathbb{Q}$ is the set of rational numbers, and $<$ is the usual ordering of $\mathbb{Q}$.

Remark 5.2.6. It can be shown that, up to isomorphism, $(\mathbb{Q}, <)$ is the unique countable dense linear ordering without end points. (This is proved by a back-and-forth argument.) Hence, if we let DLO be the theory of dense linear ordering without end points, DLO is $\aleph_0$-categorical. It follows by Vaught's Test that DLO is complete.

Example 5.2.7 (graphs)**.** The *language of graphs* consists of a binary predicate, R, plus the equality predicate, $=$. The *theory of graphs* consists of the equality axioms plus $\forall x\, \forall y\, (Rxy \Leftrightarrow Ryx)$ and $\forall x\, \neg\, Rxx$. A *graph* is a model of the theory of graphs.

Thus a graph is essentially an ordered pair $G = (V_G, R_G)$, where V_G is a nonempty set and R_G is a symmetric, irreflexive relation on V_G. The elements of V_G are called *vertices*. Two vertices $a, b \in V_G$ are said to be *adjacent* if $\langle a, b\rangle \in R_G$. A *path* from a to b is a finite sequence of pairwise distinct vertices $a = v_0, v_1, \ldots, v_n = b$ such that $a = v_0$ is adjacent to v_1, v_1 is adjacent to v_2, $\ldots$, v_{n-1} is adjacent to $v_n = b$. G is said to be *connected* if for all $a, b \in V_G$ there exists a path from a to b. Equivalently, G is connected if and only if, for all partitions of V_G into two disjoint nonempty sets X and Y, there exist $a \in X$ and $b \in Y$ such that a and b are adjacent.

Exercise 5.2.8. Show that there exist graphs G_1 and G_2 such that G_1 is connected, G_1 is elementarily equivalent to G_2, yet G_2 is not connected. Deduce that there is no theory T in the language of graphs such that the models of T are exactly the connected graphs.

Solution. If a and b are two vertices in a graph, we define $d(a, b)$, the *distance* from a to b, to be the smallest length of a path from a to b, or ∞ if there is no such path. Let G_1 be a graph which is connected yet contains pairs of vertices which are at distance n for arbitrarily large n. (For example, we may take $G_1 = (\mathbb{N}, R_1)$ where $R_1 = \{\langle n, n+1\rangle, \langle n+1, n\rangle \mid n \in \mathbb{N}\}$.) Let $L = \{R, =\}$ be the language of graphs, and let S be the set of L-sentences satisfied by G_1. Let $L^* = L \cup \{a, b\}$ where a, b are new constants, and let $S^* = S \cup \{A_n \mid n = 1, 2, \ldots\}$ where A_n is an L^*-sentence saying that there is no path of length n from a to b. By choosing $a, b \in G_1$ appropriately, we see that all finite subsets of S^* are normally satisfiable. Hence by the Compactness Theorem S^* is normally satisfiable, so let (G_2, a, b) be a model of S^*. Then G_2 is a graph which is elementarily equivalent to G_1, yet $a, b \in G_2$ are such that $d(a,b) = \infty$, hence G_2 is not connected.

If there were an L-theory T whose models are exactly the connected graphs, then G_1 would be a model of T but G_2 would not, contradicting the fact that G_1 and G_2 are elementarily equivalent.

Exercise 5.2.9. A graph G is said to be *random* if for all finite sets of distinct vertices $a_1, \ldots, a_m, b_1, \ldots, b_n$ there exists a vertex c such that c is adjacent to $a_1, \ldots, a_m$ and not adjacent to $b_1, \ldots, b_n$. Show that the theory of random graphs is $\aleph_0$-categorical. It follows by Vaught's Test that this theory is complete.

Solution. Let G and G' be two random graphs of cardinality $\aleph_0$, say $G = \{a_k \mid k \in \mathbb{N}\}$ and $G' = \{a'_k \mid k \in \mathbb{N}\}$. We use a back-and-forth argument to construct an isomorphism of G onto G'. At stage n of the construction, we have a finite partial isomorphism, f_n, which maps a finite subset of G isomorphically onto a finite subset of G'. Start with $f_0 = \emptyset$. Suppose we have already constructed f_n. To construct f_{n+1}, consider two cases. If n is even, let k_n be the least k such that $a_k \notin \mathrm{dom}(f_n)$, and put $c = a_{k_n}$. By randomness of G', find $c' \in G'$ such that for all $a \in \mathrm{dom}(f_n)$, c' is adjacent to $f_n(a)$ if and only if c is adjacent to a. If n is odd, let k_n be the least k such that $a'_k \notin \mathrm{ran}(f_n)$, and put $c' = a'_{k_n}$. By randomness of G, find $c \in G$ such that for all $a \in \mathrm{dom}(f_n)$, c is adjacent to a if and only if c' is adjacent to $f_n(a)$. In either case, let $f_{n+1} = f_n \cup \{(c, c')\}$. Finally, by construction, $f = \bigcup_{n=0}^{\infty} f_n$ is an isomorphism of G onto G'.

Example 5.2.10 (rings)**.** The *language of rings* consists of binary operations $+$ and $\cdot$ (addition and multiplication), a unary operation $-$ (subtraction), constants 0 and 1 (the additive and multiplicative identity elements), and a binary predicate $=$ (equality). The *theory of rings* consists of the equality axioms plus

$$\forall x \, \forall y \, \forall z \, (x + (y + z) = (x + y) + z),$$

$$\forall x \, \forall y \, (x + y = y + x),$$

$$\forall x \, (x + (-x) = 0),$$

$$\forall x \, (x + 0 = x),$$

$$\forall x \, \forall y \, \forall z \, (x \cdot (y \cdot z) = (x \cdot y) \cdot z),$$

$$\forall x \, \forall y \, \forall z \, (x \cdot (y + z) = (x \cdot y) + (x \cdot z)) \text{ (left distributivity)},$$

$$\forall x \, \forall y \, \forall z \, ((x + y) \cdot z = (x \cdot z) + (y \cdot z)) \text{ (right distributivity)},$$

$$\forall x \, (x \cdot 1 = 1 \cdot x = x),$$

$$\forall x \, (x \cdot 0 = 0 \cdot x = 0),$$

$$0 \neq 1.$$

A *ring* is a model of the theory of rings. A ring is said to be *commutative* if it satisfies the additional axiom

$$\forall x\, \forall y\, (x \cdot y = y \cdot x).$$

An example of a commutative ring is the ring of integers,

$$\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}.$$

An example of a non-commutative ring is the ring of 2×2 matrices.

Example 5.2.11 (fields)**.** A *field* is a commutative ring satisfying the additional axiom

$$\forall x\, (x \neq 0 \Rightarrow \exists y\, (x \cdot y = 1)).$$

A field is said to be *of characteristic* 0 if it satisfies

$$\underbrace{1 + \cdots + 1}_{n} \neq 0$$

for all positive integers n. Familiar fields such as the field of rational numbers, $\mathbb{Q}$, the field of real numbers, $\mathbb{R}$, and the field of complex numbers, $\mathbb{C}$, are of characteristic 0. It can be shown that if a field satisfies $\underbrace{1 + \cdots + 1}_{n} = 0$ for some positive integer n, then the least such n is a prime number, p. In this case, our field is said to be *of characteristic* p. An example of a field of characteristic p is the ring of integers modulo p.

A field F is said to be *algebraically closed* if for all nontrivial polynomials $f(z) = a_n z^n + \cdots + a_1 z + a_0$, $a_n, \ldots, a_1, a_0 \in F$, $a_n \neq 0$, $n \geq 1$, there exists $z \in F$ such that $f(z) = 0$. It is known that the complex field $\mathbb{C}$ is algebraically closed. (This theorem is known as the Fundamental Theorem of Algebra.) Note that a field is algebraically closed if and only if it satisfies the axioms

$$\forall x_0\, \forall x_1\, \cdots \forall x_n\, (x_n \neq 0 \Rightarrow \exists z\, (x_n z^n + \cdots + x_1 z + x_0 = 0))$$

for $n = 1, 2, 3, \ldots$.

Exercise 5.2.12. Show that the theory of fields of characteristic 0 is not finitely axiomatizable. Show that the theory ACF of algebraically closed fields is not finitely axiomatizable. Show that the theory ACF_0 of algebraically closed fields of characteristic 0 is not finitely axiomatizable.

Solution. Suppose that the theory of fields of characteristic 0 were finitely axiomatizable. By Compactness, the axioms would be logical consequences of the field axioms plus finitely many axioms of the form $\underbrace{1 + \cdots + 1}_{n} \neq 0$, say $n = 1, \ldots, k$. Let p be a prime number greater than k. The field of integers modulo p satisfies these axioms but is not of characteristic 0, a contradiction.

Suppose ACF or ACF_0 were finitely axiomatizable. By Compactness, the axioms would be logical consequences of the field axioms plus finitely many of the ACF_0 axioms as presented in Example 5.2.11 above. But, for any finite subset of the ACF_0 axioms, we can construct a field which satisfies these axioms yet is not algebraically closed. (The construction of such a field is perhaps somewhat delicate.) This gives a contradiction.

Exercise 5.2.13. Let L be the language of rings. Let S be the set of L-sentences which are true in the ring of integers modulo p for all but finitely many prime numbers p. Does there exist a field of characteristic 0 satisfying S? Prove your answer.

Solution. The answer is yes. Note first that, for each prime p, the ring of integers modulo p is actually a field. Hence S includes the field axioms. Let $S_0 = S \cup \{\underbrace{1 + \cdots + 1}_{n} \neq 0 \mid n = 1, 2, \ldots\}$. Any finite subset of S_0 is normally satisfiable, e.g., in the integers modulo p for all sufficiently large primes p. By the Compactness Theorem for normal satisfiability (Corollary 4.1.9), it follows that S_0 is normally satisfiable. The normal structures which satisfy S_0 are fields of characteristic 0 satisfying S.

Remark 5.2.14. Let ACF_0 be the theory of algebraically closed fields of characteristic 0. For each prime number p, let ACF_p be the theory of algebraically closed fields of characteristic p. It can be shown that the theories ACF_0 and ACF_p are κ-categorical for all uncountable cardinals κ. It follows by Vaught's Test that these theories are complete.

Example 5.2.15 (vector spaces)**.** The *language of vector spaces* is a 2-sorted language with sorts σ and τ, denoting *scalars* and *vectors* respectively. For the scalars we have binary operations $+$ and $\cdot$ of type (σ, σ, σ), a unary operation $-$ of type (σ, σ), constants 0 and 1 of type σ, and an equality predicate $=$ of type (σ, σ). For the vectors we have a binary operation $+$ of type (τ, τ, τ), a unary operation $-$ of type (τ, τ), a constant 0 of type τ, and an equality predicate $=$ of type (τ, τ). In addition we have a binary operation $\cdot$ of "mixed" type (σ, τ, τ), denoting scalar multiplication.

The *theory of vector spaces* consists of the field axioms for scalars, the Abelian group axioms for vectors, and the axioms

$$\forall x^\sigma \, \forall v^\tau \, \forall w^\tau \, (x \cdot (v + w) = (x \cdot v) + (x \cdot w)),$$

$$\forall x^\sigma \forall v^\tau \, (x \cdot (-v) = -(x \cdot v)),$$

$$\forall x^\sigma \, (x \cdot 0^\tau = 0),$$

$$\forall x^\sigma \, \forall y^\sigma \, \forall v^\tau \, ((x + y) \cdot v = (x \cdot v) + (y \cdot v)),$$

$$\forall x^\sigma \forall v^\tau \, ((-x) \cdot v = -(x \cdot v)),$$

$$\forall x^\sigma \, \forall y^\sigma \, \forall v^\tau \, ((x \cdot y) \cdot v = x \cdot (y \cdot v)),$$

$$\forall v^\tau \, (1 \cdot v = v),$$

$$\forall v^\tau \, (0 \cdot v = 0)$$

for scalar multiplication. A *vector space* is a model of these axioms.

If V is a vector space, a set of vectors S in V is said to *span* V if every vector v in V can be written as a *linear combination* of vectors in S, i.e., $v =$

$a_1 \cdot v_1 + \cdots + a_n \cdot v_n$ for some $v_1, \ldots, v_n \in S$ and scalars $a_1, \ldots, a_n$. An important invariant of a vector space is its *dimension*, i.e., the minimum cardinality of a spanning set. It can be shown that, up to isomorphism over a field F, the unique vector space over F of dimension κ is the familiar space $\bigoplus_{i<\kappa} F$. Here the vectors are sequences $\langle a_i \mid i < \kappa \rangle$, with $a_i \in F$ for all i, and $a_i = 0$ for all but finitely many i. Vector addition is given by

$$\langle a_i \mid i < \kappa \rangle + \langle b_i \mid i < \kappa \rangle = \langle a_i + b_i \mid i < \kappa \rangle,$$

and scalar multiplication is given by

$$c \cdot \langle a_i \mid i < \kappa \rangle = \langle c \cdot a_i \mid i < \kappa \rangle.$$

Example 5.2.16 (ordered algebraic structures)**.** The *language of ordered rings* consists of the language of rings $+, \cdot, -, 0, 1, =$, together with $<$. The *ordered field axioms* consist of the field axioms, plus the linear ordering axioms, plus

$$\forall x \,\forall y \,\forall z \,(x < y \Leftrightarrow x + z < y + z),$$

$$\forall x \,\forall y \,((x > 0 \wedge y > 0) \Rightarrow x \cdot y > 0).$$

An *ordered field* is a model of these axioms. An example of an ordered field is the field of rational numbers $(\mathbb{Q}, <)$ with its usual ordering.

An ordered field $(F, <)$ is said to be *real-closed* if for all polynomials $f(x) = a_n x^n + \cdots + a_1 x + a_0$, $a_n, \ldots, a_1, a_0 \in F$, and for all $b, c \in F$, if $f(b) < 0 < f(c)$ then there exists $x \in F$ between b and c such that $f(x) = 0$. Clearly the ordered field of real numbers $(\mathbb{R}, <)$ is real-closed. Note that an ordered field is real-closed if and only if it satisfies the axioms

$$\begin{aligned}
&\forall u \,\forall v \,\forall w_0 \,\forall w_1 \,\cdots \forall w_n \\
&((u < v \wedge w_n u^n + \cdots + w_1 u + w_0 < 0 < w_n v^n + \cdots + w_1 v + w_0) \\
&\Rightarrow \exists x \,(u < x < v \wedge w_n x^n + \cdots + w_1 x + w_0 = 0))
\end{aligned}$$

for $n = 1, 2, 3, \ldots$.

Remark 5.2.17 (elimination of quantifiers)**.** Let RCOF be the theory of real-closed ordered fields. A famous and important theorem of Tarski says that RCOF is complete. This holds despite the fact that RCOF is not κ-categorical for any κ.

Tarski's method of proof is as follows. A theory $T = (L, S)$ is said to admit *elimination of quantifiers* if for all L-formulas A there exists a quantifier-free L-formula A^* such that $T \vdash$ the universal closure of $A \Leftrightarrow A^*$. Tarski uses algebraic methods to show that the theory RCOF admits elimination of quantifiers. For example, the formula $\exists x \,(ax^2 + bx + c = 0)$ is equivalent over RCOF to the quantifier-free formula $(a = b = c = 0) \vee (a = 0 \neq b) \vee (a \neq 0 \leq b^2 - 4ac)$.

As a special case of quantifier elimination, we have that each sentence in the language of RCOF is equivalent over RCOF to a quantifier-free sentence. On the other hand, it is evident that the quantifier-free sentences of the language of RCOF are of a very simple nature, e.g., $1 + 0 = 1 \wedge (1 \cdot 1) + 1 < 1 + 1 + 1$. Since the truth values of such sentences are decided by the axioms of RCOF, it follows that RCOF is complete.

Exercise 5.2.18. Which of the following theories are complete? Justify your answers.

1. The theory of dense linear orderings with end points.

2. The theory of fields of characteristic 0.

3. The theory of infinite, torsion-free, Abelian groups.

4. The theory of finite-dimensional vector spaces over a field of 5 elements.

5. The theory of infinite-dimensional vector spaces over a field of 5 elements.

Solution.

1. As noted in Remark 5.2.6, the theory of dense linear orderings without end points is $\aleph_0$-categorical. It follows immediately that the theory of dense linear orderings *with* end points is $\aleph_0$-categorical. Hence, by Vaught's Test, each of these theories is complete.

2. The fields $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{C}$ are of characteristic 0, yet they are not elementarily equivalent, as can be seen by considering the sentences $\exists x\,(x \cdot x = 1 + 1)$ and $\exists z\,(z \cdot z = -1)$. Therefore, the theory of fields of characteristic 0 is incomplete.

3. The additive groups of $\mathbb{Z}$ and $\mathbb{Q}$ are infinite, Abelian, and torsion-free, yet they are not elementary equivalent, as can be seen by considering the sentence $\exists x\,(x + x = 1)$. Therefore, the theory of infinite, Abelian, torsion-free groups is incomplete.

4. The vector spaces of dimension $n = 0, 1, 2, \ldots$ over a particular finite field F are not elementarily equivalent, because they have different finite cardinalities $1, q, q^2, \ldots$ where q is the cardinality of F. Therefore, the theory of finite-dimensional vector spaces over F is incomplete.

5. As noted in Example 5.2.15, any two vector spaces over the same field of the same dimension are isomorphic. On the other hand, for any infinite cardinal number κ, any vector space of cardinality κ over a finite field is of dimension κ. Combining these facts, we see that for any particular finite field F, the theory T_F of infinite-dimensional vector spaces over F is κ-categorical. Hence, by Vaught's Test, T_F is complete.

5.3 Definability over a Model

Definition 5.3.1 (explicit definability)**.** Let L be a language, let M be an L-structure, and let R be an n-ary relation on U_M. We say that R is *explicitly definable over* M, or just *definable over* M, if there exists an L-formula D with n free variables $x_1, \ldots, x_n$ such that

$$R = \{\langle a_1, \ldots, a_n \rangle \mid M \models D[x_1/a_1, \ldots, x_n/a_n]\}.$$

Example 5.3.2. Consider the binary relation $< \, = \{\langle a, b\rangle \in \mathbb{R}^2 \mid a < b\}$ on the set $\mathbb{R}$ of real numbers. Viewing $\mathbb{R}$ as a commutative ring, we see that for all $a, b \in \mathbb{R}$, $a < b$ if and only if $\mathbb{R} \models \exists x\, (x \neq 0 \wedge a + x^2 = b)$. Thus $<$ is explicitly definable over the commutative ring $\mathbb{R} = (\mathbb{R}, +, -, \cdot, 0, 1, =)$ by the formula $\exists z\, (z \neq 0 \wedge x + z^2 = y)$ with free variables x, y.

On the other hand, in view of Tarski's theorem on elimination of quantifiers for RCOF (see Remark 5.2.17), any subset of $\mathbb{R}$ which is definable over $\mathbb{R}$ consists of a union of finitely many points and intervals. From this it follows that, for example, the set of integers is not definable over $\mathbb{R}$.

Example 5.3.3. Let $\mathbb{N} = (\mathbb{N}, +, \cdot, 0, 1, <, =)$ be the natural number system. It can be shown that the class of relations which are explicitly definable over $\mathbb{N}$ includes all relations which are computable in the sense of Turing. In particular, the 3-ary relation $\{\langle m, n, k\rangle \in \mathbb{N}^3 \mid m = n^k\}$ is definable over $\mathbb{N}$, as we now show.

Theorem 5.3.4. The exponential function $(n, k) \mapsto n^k$ is explicitly definable over the natural number system.

Proof. The proof uses some number-theoretic lemmas, as follows.

Lemma 5.3.5 (the Chinese Remainder Theorem)**.** Let $m_1, \ldots, m_k$ be pairwise relatively prime. Given $r_1, \ldots, r_k$, we can find r such that $r \equiv r_i \bmod m_i$ for all $i = 1, \ldots, k$.

Proof. We omit the proof of this well-known result. $\square$

Lemma 5.3.6. For each $k \in \mathbb{N}$ we can find infinitely many $a \in \mathbb{N}$ such that the integers $a + 1, 2a + 1, \ldots, ka + 1$ are pairwise relatively prime.

Proof. Let a be any multiple of $k!$. Suppose p is a prime number which divides both $ai + 1$ and $aj + 1$ where $1 \leq i < j \leq k$. Since p divides $ai + 1$ and a is a multiple of $k!$, we clearly have $p > k$. On the other hand, p divides $a(j - i)$, and $j - i < k$, hence p divides a, a contradiction since p divides $ai + 1$. $\square$

Definition 5.3.7 (Gödel's beta-function)**.** We define $\beta(a, r, i) =$ the remainder of r upon division by $a \cdot (i+1) + 1$. Note that the β-function is explicitly definable over the natural number system.

Lemma 5.3.8. Given a finite sequence of natural numbers $n_0, n_1, \ldots, n_k$, we can find natural numbers a, r such that $\beta(a, r, i) = n_i$ for all $i = 0, 1, \ldots, k$.

Proof. Let $a \geq \max\{n_i \mid i \leq k\}$ be such that $a \cdot (i+1)+1$, $i \leq k$, are pairwise relatively prime. By the Chinese Remainder Theorem, we can find r such that $r \equiv n_i \bmod a \cdot (i+1)+1$ for all $i \leq k$. Since $0 \leq n_i \leq a < a \cdot (i+1)+1$, it follows that $\beta(a,r,i) = n_i$. □

Now, to prove Theorem 5.3.4, note that $m = n^k$ holds if and only if $\exists a\, \exists r\, (\beta(a,r,0) = 1 \wedge \beta(a,r,k) = m \wedge \forall i\, (i < k \Rightarrow \beta(a,r,i) \cdot n = \beta(a,r,i+1)))$. □

Exercise 5.3.9. Let p_n be the nth prime number. Thus $p_0 = 2$, $p_1 = 3$, $p_2 = 5$, $p_3 = 7$, $p_4 = 11$, etc. Show that the function $k \mapsto p_k$ is explicitly definable over the natural number system $\mathbb{N}$. (This means, show that the binary relation $\{\langle k, p_k\rangle \mid k \in \mathbb{N}\}$ is explicitly definable over $\mathbb{N}$.)

Solution. Let A be a formula expressing that x is prime, for example

$$1 < x \wedge \neg \exists u\, \exists v\, (u < x \wedge v < x \wedge u \cdot v = x).$$

We then have $p_k = n$ if and only if $\exists a\, \exists r\, (B \wedge C)$, where B is

$$\beta(a,r,k) = n \wedge \forall i\, (i < k \Rightarrow \beta(a,r,i) < \beta(a,r,i+1))$$

and C is

$$\forall x\, (x \leq n \Rightarrow (A \Leftrightarrow \exists i\, (i \leq k \wedge \beta(a,r,i) = x))).$$

We now turn to implicit definability.

Definition 5.3.10 (implicit definability)**.** Let L be a language, let M be an L-structure, and let R be an n-ary relation on U_M. We say that R is *implicitly definable over* M if there exists a sentence D^* in the language $L^* = L \cup \{\underline{R}\}$ with an additional n-ary predicate $\underline{R}$, such that for all n-ary relations R' on U_M, $(M, R') \models D^*$ if and only if $R' = R$.

Example 5.3.11. Let $\mathbb{R} = (\mathbb{R}, +, -, \cdot, 0, 1, <, =)$ be the ordered field of real numbers. Consider the set of integers, $\mathbb{Z} \subset \mathbb{R}$. As noted above, $\mathbb{Z}$ is not explicitly definable over $\mathbb{R}$. However, $\mathbb{Z}$ is implicitly definable over $\mathbb{R}$, by the following sentence with an additional unary predicate Z:

$$Z0 \wedge Z1 \wedge (\neg \exists x\, (Zx \wedge 0 < x < 1)) \wedge (\forall x\, (Zx \Leftrightarrow Z(x+1))).$$

Example 5.3.12. It can be shown that there exists a subset of $\mathbb{N}$ which is implicitly definable over the natural number system $\mathbb{N} = (\mathbb{N}, +, \cdot, 0, 1, <, =)$ but is not explicitly definable over $\mathbb{N}$. See Remark 6.4.6 and Exercise 6.4.7 below.

Definition 5.3.13 (automorphisms)**.** Let M be a structure. An *automorphism* of M is an isomorphism of M onto itself. An n-ary relation R on U_M is said to be *invariant* if

$$R = \{\langle f(a_1), \ldots, f(a_n)\rangle \mid \langle a_1, \ldots, a_n\rangle \in R\}$$

for all automorphisms f of M.

Exercise 5.3.14. Let M be a structure. Show that any relation which is explicitly definable over M is implicitly definable over M. Show that any relation which is implicitly definable over M is invariant under all automorphisms of M. Give counterexamples showing that the converses of these assertions fail in general.

Solution. Let R be an n-ary relation over U_M. If R is explicitly defined over M by a formula D with free variables $x_1, \ldots, x_n$, then R is implicitly defined over M by the $L \cup \{\underline{R}\}$-sentence $\forall x_1 \cdots \forall x_n\, (\underline{R} x_1 \cdots x_n \Leftrightarrow D)$. Now suppose R is implicitly defined over M by an $L \cup \{\underline{R}\}$-sentence D^*. Let f be an automorphism of M, and put $R' = \{\langle f(a_1), \ldots, f(a_n)\rangle \mid \langle a_1, \ldots, a_n\rangle \in R\}$. Then f is an isomorphism of (M, R) onto (M, R'). Since $(M, R) \models D^*$, it follows by Theorem 2.2.6 that $(M, R') \models D^*$. Hence $R' = R$, i.e., R is invariant under f.

Consider the ordered field $\mathbb{R}$ of real numbers. Example 5.3.11 shows that the subset $\mathbb{Z}$ of $\mathbb{R}$ is implicitly definable over $\mathbb{R}$ but not explicitly definable over $\mathbb{R}$. Since there are only countably many sentences, only countably many subsets of $\mathbb{R}$ are implicitly definable over $\mathbb{R}$. However, *all* subsets of $\mathbb{R}$ are invariant under automorphisms of $\mathbb{R}$, inasmuch as $\mathbb{R}$ has no automorphisms except the identity.

Exercise 5.3.15. Let $\mathbb{R} = (\mathbb{R}, +, -, \cdot, 0, 1, <, =)$ be the ordered field of real numbers. Show that the relations $y = e^x$ and $y = \sin x$ are implicitly definable over $\mathbb{R}$. It can be shown that these relations are not explicitly definable over $\mathbb{R}$. (Hint: The relations $y = e^x$ and $y = \sin x$ are implicitly defined by differential equations which can be expressed as formulas of the predicate calculus, using the ϵ-δ-method.)

Exercise 5.3.16. Show that if U_M is finite, then any relation which is implicitly definable over M is explicitly definable over M.

Solution. Let us say that $\langle a_1, \ldots, a_n\rangle, \langle b_1, \ldots, b_n\rangle \in (U_M)^n$ are of the same *n-type* if $(M, a_1, \ldots, a_n)$ is elementarily equivalent to $(M, b_1, \ldots, b_n)$. Thus $(U_M)^n$ is partitioned into equivalence classes, the n-types. Since U_M is finite, there are only finitely many n-types, and each n-type is explicitly definable over M. Moreover, by Theorem 2.2.6 and Exercise 4.1.12, the n-types are just the orbits of $(U_M)^n$ under the automorphism group of M. If $R \subseteq (U_M)^n$ is implicitly definable over M, then R is invariant under automorphisms of M, hence R is the union of some of the n-types, hence R is explicitly definable over M.

Exercise 5.3.17. (In this exercise we assume familiarity with saturated models.) Show that if a structure M is saturated, then any relation which is implicitly definable over M is explicitly definable over M.

Exercise 5.3.18. Let G be a group which has infinitely many distinct subgroups. Prove that there exists a countable group G' such that

1. G' is elementarily equivalent to G, and

2. G' has a subgroup which is not explicitly definable over G'.

Solution. Let S be the set of sentences which are true in G. Introduce a new unary predicate P, and let S' consist of S plus the sentences $P1$ and $\forall x\,(Px \Rightarrow Px^{-1})$ and $\forall x\,\forall y\,((Px \wedge Py) \Rightarrow P(x \cdot y))$ and $\neg\,\forall x\,(Px \Leftrightarrow D)$ where D is any formula with x as its only free variable. Since G has infinitely many subgroups, each finite subset of S' is normally satisfiable in G by letting P be an appropriately chosen subgroup of G. It follows by Compactness plus Löwenheim/Skolem that we can find a countable normal structure (G', P') satisfying S'. Then G' is a countable group which is elementarily equivalent to G, and P' is a subgroup of G' which is not explicitly definable over G'.

Remark 5.3.19. In this section we have considered explicit and implicit definability over a *model*, M. We have given examples showing that, in general, implicit definability over M does not imply explicit definability over M.

In Sections 5.4 and 5.8 below, we shall consider the related but more restrictive notions of explicit and implicit definability over a *theory*, T. It will be obvious that explicit definability over T implies explicit definability over any model of T, and implicit definability over T implies implicit definability over any model of T. A pleasant surprise is that explicit definability over T is equivalent to implicit definability over T. This is the content of Beth's Definability Theorem, Theorem 5.8.2 below.

5.4 Definitional Extensions of Theories

We now show how theories can be usefully extended by adding new predicates and operations which are explicitly definable in terms of old predicates and operations. This method of extending a theory is known as *definitional extension*.

In this section we are considering only explicit definitional extensions. Later, in Section 5.8, we shall consider implicit definitional extensions. It will turn out that, in principle, an implicit definitional extension of a theory is always equivalent to an explicit definitional extension of the same theory.

Definition 5.4.1 (defining a new predicate)**.** Let $T = (L, S)$ be a theory, and let D be an L-formula with free variables $x_1, \ldots, x_n$. Introduce a new n-ary predicate P, and let $T' = (L', S')$ where $L' = L \cup \{P\}$ and $S' = S \cup \{\forall x_1 \cdots \forall x_n\,(Px_1 \cdots x_n \Leftrightarrow D)\}$.

Definition 5.4.2 (defining a new operation)**.** Let $T = (L, S)$ be a theory with an identity predicate I. Let D be an L-formula with free variables $x_1, \ldots, x_n, y$ such that

$$T \vdash \forall x_1 \cdots \forall x_n\,(\exists \text{ exactly one } y)\,D,$$

i.e.,

$$T \vdash \forall x_1 \cdots \forall x_n\,\exists z\,\forall y\,(Iyz \Leftrightarrow D)$$

where z is a new variable. Introduce a new n-ary operation f, and let $T' = (L', S')$ where $L' = L \cup \{f\}$ and $S' = S \cup \{\forall x_1 \cdots \forall x_n\,\forall y\,(Ifx_1 \ldots x_n y \Leftrightarrow D)\}$.

Remark 5.4.3. In Definition 5.4.2 above, we have assumed for simplicity that T is one-sorted. In case T is many-sorted, we make the obvious modifications. Namely, letting $\sigma_1, \ldots, \sigma_n, \tau$ be the sorts of the variables $x_1, \ldots, x_n, y$ respectively, we require that z is a new variable of sort τ, I is an identity predicate of sort τ, and f is an operation of type $(\sigma_1, \ldots, \sigma_n, \tau)$.

Remark 5.4.4. We are going to prove that these extensions of T are "trivial" or "harmless" or "inessential", in the sense that each formula in the extended language $L \cup \{P\}$ or $L \cup \{f\}$ can be straightforwardly translated into an equivalent formula of the original language L. See Theorem 5.4.10 below.

Lemma 5.4.5. Let $T = (L, S)$ and $T' = (L', S')$ be as in Definition 5.4.1 or 5.4.2. Then for all L-sentences A we have $T' \vdash A$ if and only if $T \vdash A$.

Proof. It suffices to show that, for each model M of T, there exists a model M' of T' such that $M = M' \restriction L$, i.e., M is the *reduct* of M' to L. In the case of Definition 5.4.1, let $M' = (M, P_{M'})$ where

$$P_{M'} = \{\langle a_1, \ldots, a_n \rangle \mid M \models D[x_1/a_1, \ldots, x_n/a_n]\}.$$

In the case of Definition 5.4.2, let $M' = (M, f_{M'})$ where $f_{M'}(a_1, \ldots, a_n) =$ the unique b such that $M \models D[x_1/a_1, \ldots, x_n/a_n, y/b]$. Clearly M' is as desired. $\square$

Definition 5.4.6 (translation)**.** Let $T = (L, S)$ and $T' = (L', S')$ be as in Definition 5.4.1 or 5.4.2. To each L'-formula A we associate an L-formula A', the *translation* of A into L. In order to define A', we first modify D, replacing the free and bound variables of D by new variables which do not occur in A. (Compare Definition 2.4.9.) In the case of Definition 5.4.1, where we are adding a new predicate P, we obtain A' from A by replacing each atomic formula of the form $Pt_1 \ldots t_n$ by $D[x_1/t_1, \ldots, x_n/t_n]$.

In the case of Definition 5.4.2, where we are adding a new operation f, the translation is more complicated. For non-atomic A we obtain A' by induction on the degree of A, putting $(\neg A)' = \neg A'$, $(A \wedge B)' = A' \wedge B'$, $(\forall x\, A)' = \forall x\, A'$, etc. For atomic A we obtain A' by induction on the number of occurrences of f in A. If there are no occurrences of f in A, let A' be just A. Otherwise, write A in the form $B[w/ft_1 \cdots t_n]$ where w is a new variable and f does not occur in $t_1, \ldots, t_n$, and let A' be

$$\exists w\, (D[x_1/t_1, \ldots, x_n/t_n, y/w] \wedge B')$$

or equivalently

$$\forall w\, (D[x_1/t_1, \ldots, x_n/t_n, y/w] \Rightarrow B').$$

Lemma 5.4.7. Let T and T' be as in Definition 5.4.1 or 5.4.2, and let $A \mapsto A'$ be our translation of L'-formulas to L-formulas as in Definition 5.4.6. Then:

1. A' has the same free variables as A and is equivalent to it over T', i.e., $T' \vdash A \Leftrightarrow A'$.

2. Propositional connectives and quantifiers are respected, i.e., we have $(\neg A)' \equiv \neg A'$, $(A \wedge B)' \equiv A' \wedge B'$, $(\forall x\, A)' \equiv \forall x\, A'$, etc.

3. If A happens to be an L-formula, then A' is just A.

Consequently, for all L'-sentences A we have $T' \vdash A$ if and only if $T \vdash A'$.

Proof. The first part is straightforward. For the last part, since $T' \vdash A \Leftrightarrow A'$, we obviously have $T' \vdash A$ if and only if $T' \vdash A'$. But then, since A' is an L-sentence, it follows by Lemma 5.4.5 that $T' \vdash A$ if and only if $T \vdash A'$. $\square$

Definition 5.4.8 (definitional extensions)**.** Let T be a theory. A *definitional extension* of T is a theory T^* which is a union of sequences of theories $T_0, T_1, \ldots, T_k$ beginning with $T_0 = T$ such that each T_{i+1} is obtained by extending T_i as in Definition 5.4.1 or 5.4.2.

Definition 5.4.9 (conservative extensions)**.** Let $T = (L, S)$ and $T^* = (L^*, S^*)$ be theories such that $L^* \supseteq L$. We say that T^* is a *conservative extension* of T if, for all L-sentences A, $T^* \vdash A$ if and only if $T \vdash A$.

Theorem 5.4.10. Let T^* be a definitional extension of T. Then T^* is a conservative extension of T. Moreover, there is a straightforward translation $A \mapsto A^*$ of L^*-formulas to L-formulas, with the following properties.

1. A^* has the same free variables as A and is equivalent to it over T^*, i.e., $T^* \vdash A \Leftrightarrow A^*$.

2. Propositional connectives and quantifiers are respected, i.e., we have $(\neg A)^* \equiv \neg A^*$, $(A \wedge B)^* \equiv A^* \wedge B^*$, $(\forall x\, A)^* \equiv \forall x\, A^*$, etc.

3. If A happens to be an L-formula, then A^* is just A.

Consequently, for all L^*-sentences A, we have

$$T^* \vdash A \quad \text{if and only if} \quad T \vdash A^*.$$

Proof. This is clear from Lemmas 5.4.5 and 5.4.7. $\square$

Remark 5.4.11. In practice, when it comes to working with specific theories T, the technique of definitional extensions is very useful. This is because formulas written in the extended language L^* tend to be much shorter than their translations in L. This is particularly important for the foundational theories Z_1, Z_2, ..., Z_n, ..., Z_∞, ZC, ZFC which are discussed in Sections 5.5 and 5.6 below.

Exercise 5.4.12. Show that for any theory T there is a definitional extension of T which admits elimination of quantifiers.

Solution. Let L be the language of T. For each L-formula A with free variables $x_1, \ldots, x_n$ introduce a new n-ary predicate P_A and a new axiom

$$\forall x_1 \cdots \forall x_n\, (P_A x_1 \cdots x_n \Leftrightarrow A).$$

The resulting theory admits elimination of quantifiers.

Exercise 5.4.13. Let $T = (L, S)$ be a theory. Let A be an L-formula with free variables $x_1, \ldots, x_n, y$. Let $T' = (L', S')$ where L' consists of L plus a new n-ary operation f, and S' consists of S plus $\forall x_1 \cdots \forall x_n\,(A[y/fx_1 \cdots x_n] \Leftrightarrow \exists y\, A)$. Show that T' is a conservative extension of T.

Solution. Let B be an L-sentence such that $T' \vdash B$. We must show that $T \vdash B$. If $T \nvdash B$, let M be a model of $T \cup \{\neg B\}$. Let $M' = (M, f_{M'})$ where $f_{M'}$ is an n-ary function on U_M with the following property: for all $a_1, \ldots, a_n \in U_M$ such that $M \models \exists y\, A[x_1/a_1, \ldots, x_n/a_n]$, we have $M \models A[x_1/a_1, \ldots, x_n/a_n, y/b]$ where $b = f_{M'}(a_1, \ldots, a_n)$. (Note that $f_{M'}$ is not necessarily unique and is not necessarily definable over M. The existence of an $f_{M'}$ with the desired property follows from the axiom of choice.) Clearly M' is a model of $T' \cup \{\neg B\}$, hence $T' \nvdash B$, a contradiction.

5.5 Foundational Theories

In this section and the next, we present several specific theories which are of significance in the foundations of mathematics. We point out that these theories are, in certain senses, "almost complete" or "practically complete." On the other hand, we shall see in Chapter 6 that these theories are not complete.

Definition 5.5.1 (first-order arithmetic)**.** One of the most famous and important foundational theories is *first-order arithmetic*, Z_1, also known as *Peano Arithmetic*, PA. The *language of first-order arithmetic* is $L_1 = \{+, \cdot, 0, S, =\}$ where $+$ and $\cdot$ are binary operations, S is a unary operation, 0 is a constant, and $=$ is the equality predicate. Let Q be the finitely axiomatizable L_1-theory with axioms

$$\forall x\,(Sx \neq 0),$$

$$\forall x\,\forall y\,(Sx = Sy \Rightarrow x = y),$$

$$\forall x\,(x + 0 = x),$$

$$\forall x\,\forall y\,(x + Sy = S(x + y)),$$

$$\forall x\,(x \cdot 0 = 0),$$

$$\forall x\,\forall y\,(x \cdot Sy = (x \cdot y) + x).$$

The axioms of Z_1 consist of the axioms of Q plus the *induction scheme*, i.e., the universal closure of

$$(A[x/0] \wedge \forall x\,(A \Rightarrow A[x/Sx])) \Rightarrow \forall x\, A$$

where A is any L_1-formula.

Note that the induction scheme consists of an infinite set of axioms. It can be shown that Z_1 is not finitely axiomatizable.

Remark 5.5.2 (practical completeness of first-order arithmetic)**.** The intended model of Z_1 is the natural number system

$$\mathbb{N} = (\mathbb{N}, +, \cdot, 0, S, =)$$

where $\mathbb{N} = \{0, 1, 2, \ldots\}$ and $+, \cdot, 0, =$ are as expected, and S is the *successor function*, $S(n) = n + 1$ for all $n \in \mathbb{N}$. Obviously the axioms of Z_1 are true in this model. The idea behind the axioms of Z_1 is that we are trying to write down a set S_1 of L_1-sentences with the property that, for all L_1-sentences A, $S_1 \vdash A$ if and only if $\mathbb{N} \models A$. In particular, the axioms of Z_1 are intended to be complete. Unfortunately this intention cannot be fulfilled, as shown by Gödel's First Incompleteness Theorem (see Chapter 6). However, it is known that Z_1 "almost" fulfills the intention. This is because it has been found that, in practice, all or most of the theorems of number theory which can be written in (definitional extensions of) Z_1 are either provable or refutable in Z_1. Exceptions are known, but the exceptions are obscure and marginal. In this sense, we can say that Z_1 is "practically complete."

Remark 5.5.3 (definitional extensions of first-order arithmetic)**.** We have seen in Theorem 5.3.4 and Exercise 5.3.9 that the exponential function $(n, k) \mapsto n^k$ and the "kth prime" function $k \mapsto p_k$ are definable over $\mathbb{N}$, the intended model of Z_1. It can be shown that the same definitions work abstractly over the theory Z_1. Thus we have a definitional extension of Z_1 in which basic properties such as $\forall m \, \forall n \, \forall k \, (m^k n^k = (mn)^k)$ and $\forall n \, \forall i \, \forall j \, (n^i n^j = n^{i+j})$ can be stated and proved. Similarly, many other number-theoretical operations can be introduced, and their properties proved, in definitional extensions of Z_1. Our remarks above concerning practical completeness of Z_1 apply all the more to these definitional extensions of Z_1.

Exercise 5.5.4. Show that the commutative laws $\forall x \, \forall y \, (x + y = y + x)$ and $\forall x \, \forall y \, (x \cdot y = y \cdot x)$ are theorems of Z_1. Similarly it can be shown that the associative laws $\forall x \, \forall y \, \forall z \, (x + (y + z) = (x + y) + z)$ and $\forall x \, \forall y \, \forall z \, (x \cdot (y \cdot z) = (x \cdot y) \cdot z)$ and the distributive law $\forall x \, \forall y \, \forall z \, (x \cdot (y + z) = (x \cdot y) + (x \cdot z))$ are theorems of Z_1.

Definition 5.5.5 (second-order arithmetic)**.** Another important foundational theory is *second-order arithmetic*, Z_2. See also my book [3].

The *language of* Z_2 is a 2-sorted language, L_2, with sorts σ and τ designating *numbers* and *sets* respectively. The *number variables* $x^\sigma, y^\sigma, \ldots$ are written as $i, j, k, l, m, n, \ldots$, while the *set variables* $x^\tau, y^\tau, \ldots$ are written as $X, Y, Z, \ldots$.

The predicates and operations of L_2 are $+, \cdot, 0, S, =, \in$. Here $+$ and $\cdot$ are binary number operations, S is a unary number operation, and 0 is a numerical constant. Thus $+$ and $\cdot$ are of type (σ, σ, σ), S is of type (σ, σ), and 0 is of type σ. In addition, the numerical equality predicate $=$ is a binary predicate of type (σ, σ), and the membership predicate $\in$ is a binary predicate of "mixed" type (σ, τ).

We identify the variables $x, y, \ldots$ of L_1 with the number variables $m, n, \ldots$ of L_2. Also, we identify the operations and predicates $+, \cdot, 0, S, =$ of L_1 with the $+, \cdot, 0, S, =$ of L_2. Thus L_1 is a sublanguage of L_2.

The axioms of Z_2 consist of Q (Definition 5.5.1), i.e.,

$$\forall m\,(Sm \neq 0),$$

$$\forall m\,\forall n\,(Sm = Sn \Rightarrow m = n),$$

$$\forall m\,(m + 0 = m),$$

$$\forall m\,\forall n\,(m + Sn = S(m + n)),$$

$$\forall m\,(m \cdot 0 = 0),$$

$$\forall m\,\forall n\,(m \cdot Sn = (m \cdot n) + m),$$

plus the *induction axiom*

$$\forall X\,((0 \in X \wedge \forall n\,(n \in X \Rightarrow Sn \in X)) \Rightarrow \forall n\,(n \in X)),$$

plus the *comprehension scheme*, i.e., the universal closure of

$$\exists X\,\forall n\,(n \in X \Leftrightarrow A)$$

where A is any L_2-formula in which X does not occur.

Note that the comprehension scheme consists of an infinite set of axioms. It can be shown that Z_2 is not finitely axiomatizable.

Remark 5.5.6 (practical completeness of second-order arithmetic)**.** The intended model of Z_2 is the 2-sorted structure

$$P(\mathbb{N}) = (\mathbb{N}, P(\mathbb{N}), +, \cdot, 0, S, =, \in)$$

where $(\mathbb{N}, +, \cdot, 0, S, =)$ is the natural number system (see Remark 5.5.2), $P(\mathbb{N})$ is the *power set* of $\mathbb{N}$, i.e., the set of all subsets of $\mathbb{N}$,

$$P(\mathbb{N}) = \{X \mid X \subseteq \mathbb{N}\},$$

and $\in$ is the *membership relation* between natural numbers and sets of natural numbers, i.e.,

$$\in \;=\; \{\langle n, X\rangle \in \mathbb{N} \times P(\mathbb{N}) \mid n \in X\}.$$

As in the case of Z_1 (compare Remark 5.5.2), it is obvious that the axioms of Z_2 are true in the 2-sorted structure $P(\mathbb{N})$, and again, the idea behind Z_2 is that we are trying to write down axioms for the complete theory of $P(\mathbb{N})$. This intention cannot be fulfilled because of the Gödel Incompleteness Theorem, but again, Z_2 is "practically complete."

Remark 5.5.7 (definitional extensions of second-order arithmetic)**.** The foundational significance of Z_2 is that, within definitional extensions of Z_2, it is possible and convenient to develop the bulk of ordinary countable and separable mathematics. This includes differential equations, analysis, functional analysis, algebra, geometry, topology, combinatorics, descriptive set theory, etc. For details of how this can be done, see my book [3].

Exercise 5.5.8. As noted in Defintion 5.5.5, L_1 is a sublanguage of L_2. Show that all of the axioms of Z_1 are theorems of Z_2. In this sense Z_1 is a subtheory of Z_2.

Solution. Recall that Q is a subtheory of Z_2. It remains to show that, for any L_1-formula A, the universal closure of $(A[n/0] \wedge \forall n\,(A \Rightarrow A[n/Sn])) \Rightarrow \forall n\, A$ is a theorem of Z_2. More generally, we shall prove this when A is any L_2-formula. Let the free variables of A be among $n, n_1, \ldots, n_k, X_1, \ldots, X_l$. Within Z_2 we reason as follows. Given $n_1, \ldots, n_k, X_1, \ldots, X_l$, we use the comprehension scheme of Z_2 to prove the existence of a set X such that $\forall n\,(n \in X \Leftrightarrow A)$. From the induction axiom of Z_2 we have $(0 \in X \wedge \forall n\,(n \in X \Rightarrow Sn \in X)) \Rightarrow \forall n\,(n \in X)$. From this it follows that $(A[n/0] \wedge \forall n\,(A \Rightarrow A[n/Sn])) \Rightarrow \forall n\, A$. Since this holds for arbitrary $n_1, \ldots, n_k, X_1, \ldots, X_l$, we obtain the universal closure.

Remark 5.5.9 (nth order arithmetic)**.** Similarly we can define *third-order arithmetic*, Z_3, a 3-sorted theory with variables intended to range over numbers, sets of numbers, and sets of sets of numbers. The intended model of Z_3 is the 3-sorted structure

$$P^2(\mathbb{N}) = (\mathbb{N}, P(\mathbb{N}), P(P(\mathbb{N})), +, \cdot, 0, S, =, \in_1, \in_2)$$

where

$$\in_1 = \{\langle n, X\rangle \in \mathbb{N} \times P(\mathbb{N}) \mid n \in X\}$$

and

$$\in_2 = \{\langle X, \mathcal{A}\rangle \in P(\mathbb{N}) \times P(P(\mathbb{N})) \mid X \in \mathcal{A}\}.$$

More generally, for each positive integer $n \geq 1$ we can define nth-order arithmetic, Z_n, an n-sorted theory whose intended model is the n-sorted structure

$$P^{n-1}(\mathbb{N}) = \underbrace{P(P \cdots (P}_{n-1}(\mathbb{N})) \cdots)$$

where P is the power set operation. Thus we have a sequence of theories

$$\mathsf{Z}_1 \subseteq \mathsf{Z}_2 \subseteq \cdots \subseteq \mathsf{Z}_n \subseteq \mathsf{Z}_{n+1} \subseteq \cdots.$$

The union $\mathsf{Z}_\infty = \bigcup_{n=1}^{\infty} \mathsf{Z}_n$ is known as *simple type theory*. This theory is historically important, because it or something like it was offered by Russell as a solution of the Russell Paradox.

5.6 Axiomatic Set Theory

We now turn to another kind of foundational theory, known as *axiomatic set theory*.

Definition 5.6.1 (the language of set theory)**.** The *language of set theory*, L_{set}, consists of two binary predicates, $\in$ and $=$, the membership predicate and the identity predicate. The variables $u, v, w, x, y, z, \ldots$ are intended to range over sets. Note that $u \in x$ means that u is an *element* of the set x, i.e., a member of x. We also use notations such as $x = \{u, v, \ldots\}$ meaning that x is a set whose elements are $u, v, \ldots$, and $x = \{u \mid \ldots\}$ meaning that x is a set whose elements are all u such that $\ldots$.

Remark 5.6.2 (pure well-founded sets)**.** In order to motivate and clarify our presentation of the axioms of set theory, we first note that the axioms are intended to apply only to sets which are *pure* and *well-founded*. A set x is said to be *pure* if all elements of x are sets, all elements of elements of x are sets, all elements of elements of elements of x are sets, etc. A set x is said to be *well-founded* if there is no infinite descending $\in$-chain

$$\cdots \in u_{n+1} \in u_n \in \cdots \in u_2 \in u_1 \in u_0 = x.$$

Remark 5.6.3. In order to state the axioms and theorems of set theory efficiently, we shall tacitly employ the technique of *definitional extensions*, which has been discussed in Section 5.4.

Definition 5.6.4 (Zermelo set theory)**.** *Zermelo set theory with the axiom of choice*, ZC, is a theory in the language L_{set} consisting of the following axioms.

1. The *axiom of extensionality*: $\forall x \, \forall y \, (x = y \Leftrightarrow \forall u \, (u \in x \Leftrightarrow u \in y))$.

   We define a binary predicate $\subseteq$ by $x \subseteq y \Leftrightarrow \forall u \, (u \in x \Rightarrow u \in y)$, i.e., x is a *subset* of y. Extensionality[1] says that $x = y$ is equivalent to $x \subseteq y \wedge y \subseteq x$.

2. The *pairing axiom*: $\forall x \, \forall y \, \exists z \, \forall u \, (u \in z \Leftrightarrow (u = x \vee u = y))$.

   By extensionality this z is unique, so we define a binary operation $\{x, y\} =$ this z. Note that $\{x, y\}$ is the *unordered pair* consisting of x and y. In addition, we define a unary operation $\{x\} = \{x, x\}$. Note that $\{x\}$ is the singleton set consisting of x.

   We also define a binary operation $(x, y) = \{\{x\}, \{x, y\}\}$, the *ordered pair* consisting of x and y. Using extensionality, we can prove the basic property

   $$\forall x \, \forall y \, \forall u \, \forall v \, ((x, y) = (u, v) \Leftrightarrow (x = u \wedge y = v)).$$

3. The *union axiom*: $\forall x \, \exists z \, \forall u \, (u \in z \Leftrightarrow \exists v \, (u \in v \wedge v \in x))$.

   By extensionality this z is unique, so we define a unary operation $\bigcup x =$ this z. Note that $\bigcup x$ is the *union* of x, i.e., the union of all of the sets which are elements of x. We also define a binary operation $x \cup y = \bigcup \{x, y\}$, the *union* of x and y.

4. The *power set axiom*: $\forall x \, \exists z \, \forall y \, (y \in z \Leftrightarrow y \subseteq x)$.

[1] Recall that L_{set} is a one-sorted language with only set variables. This fact together with the axiom of extensionality embodies our restriction to pure sets. Later we shall introduce another axiom, the axiom of foundation, which embodies our restriction to well-founded sets.

By extensionality this z is unique, so we define a unary operation $P(x) =$ this z. Note that $P(x)$ is the *power set* of x,

$$P(x) = \{y \mid y \subseteq x\},$$

the set of all subsets of x.

5. The *comprehension scheme*: the universal closure of

$$\forall x\, \exists z\, \forall u\, (u \in z \Leftrightarrow (u \in x \wedge A))$$

where A is any L_{set}-formula in which z does not occur.

The idea here is that x is a given set, and A expresses a property of elements $u \in x$. The comprehension scheme asserts the existence of a set

$$z = \{u \in x \mid A\},$$

i.e., z is a subset of x consisting of all $u \in x$ such that A holds. By extensionality, this z is unique.

Note that the comprehension scheme consists of an infinite set of axioms. It can be shown that ZC is not finitely axiomatizable.

Using comprehension, we define binary operations

(a) $x \cap y = \{u \in x \mid u \in y\}$, the *intersection* of x and y,

(b) $x \setminus y = \{u \in x \mid u \notin y\}$, the *set-theoretic difference* of x and y, and

(c) $x \times y = \{w \in P(P(x \cup y)) \mid \exists u\, \exists v\, (u \in x \wedge v \in y \wedge (u, v) = w)\}$, the *Cartesian product* of x and y.

We also define a constant $\emptyset = \{\} = x \setminus x$, the *empty set*.

We define a unary predicate $\mathrm{Fcn}(f)$ saying that f is a *function*, i.e., a set f such that

$$\forall w\, (w \in f \Rightarrow \exists x\, \exists y\, (w = (x, y)))$$

and

$$\forall x\, \forall y\, \forall z\, (((x, y) \in f \wedge (x, z) \in f) \Rightarrow y = z).$$

Using comprehension, the *domain* and *range* of f are defined as unary operations

$$\mathrm{dom}(f) = \{x \in \bigcup\bigcup f \mid \exists y\, ((x, y) \in f)\}$$

and

$$\mathrm{ran}(f) = \{y \in \bigcup\bigcup f \mid \exists x\, ((x, y) \in f)\}.$$

We also define $f(x)$, the *value* of f at $x \in \mathrm{dom}(f)$, to be the unique y such that $(x, y) \in f$.

6. The *axiom of infinity*: $\exists z\,(\emptyset \in z \wedge \forall x\,\forall y\,((x \in z \wedge y \in z) \Rightarrow x \cup \{y\} \in z))$.

   Using comprehension, we can prove the existence of a unique smallest set z as above, namely the intersection of all such sets. We define a constant HF = this z. Note that HF is an infinite set. The elements of HF are the *hereditarily finite sets*, i.e., those pure well-founded sets x such that $C(x)$ is finite. Here
$$C(x) = x \cup \bigcup x \cup \bigcup\bigcup x \cup \cdots,$$
   i.e., $C(x)$ is the set consisting of all elements of x, elements of elements of x, elements of elements of elements of x, ...

   Similarly, we can prove that there exists a unique smallest set w such that $\emptyset \in w \wedge \forall x\,(x \in w \Rightarrow x \cup \{x\} \in w)$. We define a constant ω = this w. Note that $\omega \subseteq \mathrm{HF}$. The elements of ω are just the natural numbers, inductively identified with hereditarily finite sets via $n = \{0, 1, \ldots, n-1\}$. Thus $\omega = \mathbb{N}$, the set of natural numbers, and we have $0 = \emptyset$ and, for all n, $n + 1 = n \cup \{n\}$. We define a *finite sequence* to be a function f such that $\mathrm{dom}(f) \in \omega$. We define an *infinite sequence* to be a function f such that $\mathrm{dom}(f) = \omega$.

   A set x is said to be *finite* if
$$\exists n\,\exists f\,(n \in \omega \wedge \mathrm{Fcn}(f) \wedge \mathrm{dom}(f) = n \wedge \mathrm{ran}(f) = x).$$
   We define $|x|$, the *cardinality* of x, to be the least such n. We can prove that, for all $n \in \omega$, $|n| = n$. For $m, n \in \omega$ we define
$$m + n = |(m \times \{0\}) \cup (n \times \{1\})|$$
   and $m \cdot n = |m \times n|$. On this basis, we can prove the essential properties of $+$ and $\cdot$ on ω. Moreover, from the definition of ω it follows that
$$\forall x\,((x \subseteq \omega \wedge 0 \in x \wedge \forall n\,(n \in x \Rightarrow n \cup \{n\} \in x)) \Rightarrow x = \omega).$$
   Thus we have a copy of the natural number system.

7. The *axiom of choice*:
$$\begin{aligned}&\forall f\,((\mathrm{Fcn}(f) \wedge \forall x\,(x \in \mathrm{dom}(f) \Rightarrow f(x) \neq \emptyset)) \Rightarrow\\ &\quad \exists g\,(\mathrm{Fcn}(g) \wedge \mathrm{dom}(g) = \mathrm{dom}(f) \wedge \forall x\,(x \in \mathrm{dom}(f) \Rightarrow g(x) \in f(x)))).\end{aligned}$$
   The axiom of choice says that, given an indexed family of nonempty sets, there exists a function which chooses one element from each of the sets. There is a history of controversy surrounding this axiom.

8. The *axiom of foundation*: $\forall x\,(x \neq \emptyset \Rightarrow \exists u\,(u \in x \wedge u \cap x = \emptyset))$.

   The axiom of foundation amounts to saying that all sets are well-founded. To see this, note that if there were an infinite descending $\in$-chain
$$\cdots \in u_{n+1} \in u_n \in \cdots \in u_2 \in u_1 \in u_0$$

then we would have a counterexample to the axiom of foundation, namely $x = \{u_0, u_1, \ldots, u_n, u_{n+1}, \ldots\}$. Conversely, if x were a counterexample to the axiom of foundation, i.e., $x \neq \emptyset$ and $\forall u\,(u \in x \Rightarrow u \cap x \neq \emptyset)$, then we could use the axiom of choice to obtain an infinite descending $\in$-chain $u_1 \in x$, $u_2 \in u_1 \cap x$, $u_3 \in u_2 \cap x$, ...

Remark 5.6.5 (foundational significance of set theory)**.** The significance of ZC and similar theories is that they can serve as an axiomatic, set-theoretical foundation for virtually all of mathematics. As already noted, within ZC we have the natural number system, $\mathbb{N}$. On this basis, it is possible to follow the usual Dedekind construction of the integers $\mathbb{Z}$, the rational numbers $\mathbb{Q}$, and the real numbers $\mathbb{R}$. We can also develop the theory of higher mathematical objects such as manifolds, topological spaces, operators on Banach spaces, etc., all within (definitional extensions of) ZC.

Definition 5.6.6 (Zermelo/Fraenkel set theory)**.** ZFC, *Zermelo/Fraenkel set theory with the axiom of choice*, consists of ZC, Zermelo set theory with the axiom of choice, plus

9. The *replacement scheme*: universal closure of

$$(\forall u\,(\exists \text{ exactly one } v)\,A) \Rightarrow \forall x\,\exists y\,\forall v\,(v \in y \Leftrightarrow \exists u\,(u \in x \wedge A))$$

   where A is any L_{set}-formula in which y does not occur.

   The idea here is that to each u is associated exactly one v such that A holds. Under this assumption, we assert that for all sets x there exists a set y consisting of all v associated to some $u \in x$. By extensionality, this y is unique.

Note that the replacement scheme consists of an infinite set of axioms. It can be shown that ZFC is not finitely axiomatizable.

Remark 5.6.7 (practical completeness of set theory)**.** The most obvious model of Zermelo set theory is the set

$$P^\omega(\text{HF}) = \bigcup_{n\in\omega} P^n(\text{HF}) = \bigcup_{n\in\mathbb{N}} \underbrace{P(P\cdots(P}_{n}(\text{HF}))\cdots)$$

consisting of HF plus all subsets of HF plus all subsets of subsets of HF plus .... Thus $P^\omega(\text{HF})$ is the set of all sets of finite order over HF. The axioms of ZC express evident properties of $P^\omega(\text{HF})$.

The existence of the set $P^\omega(\text{HF})$ cannot be proved in Zermelo set theory. However, the existence of $P^\omega(\text{HF})$ can be proved in Zermelo/Fraenkel set theory, as follows. Let A be a formula which associates to each $n \in \mathbb{N}$ the set $P^n(\text{HF})$. Since $\mathbb{N}$ is a set, the replacement scheme for A gives us the set $\{P^n(\text{HF}) \mid n \in \mathbb{N}\}$, and then the union axiom gives us $\bigcup\{P^n(\text{HF}) \mid n \in \mathbb{N}\} = P^\omega(\text{HF})$.

The intended model of Zermelo/Fraenkel set theory is the collection V of all pure, well-founded sets. V is also known as *the universe of set theory*. It can be shown that $V = \bigcup_\alpha P^\alpha(\mathrm{HF})$, where α ranges over transfinite ordinal numbers. Thus V is the collection of all sets of all transfinite orders over HF.

The axioms of ZFC express evident properties of V. Moreover, it has been found that ZFC is "practically complete" in the sense that all L_{set}-sentences expressing evident properties of V are provable in ZFC. At the same time, there are many interesting L_{set}-sentences, e.g., the Continuum Hypothesis, which are neither evidently true nor evidently false in V according to our current understanding, and which are known to be neither provable nor refutable in ZFC. Thus it appears that ZFC accurately reflects our current understanding of V.

5.7 Interpretability

Definition 5.7.1. Let T_1 and T_2 be theories. We say that T_1 is a *subtheory* of T_2 if the language of T_1 is included in the language of T_2 and the theorems of T_1 are included in the theorems of T_2.

Definition 5.7.2 (interpretability)**.** Let T_1 and T_2 be theories. We say that T_1 is *interpretable* in T_2 if T_1 is a subtheory of some definitional extension of T_2. Intuitively, this means that T_2 is "at least as strong as" T_1, in some abstract sense. We sometimes write $T_1 \leq T_2$ to mean that T_1 is interpretable in T_2.

Remark 5.7.3. It is straightforward to show that the interpretability relation is transitive. In other words, $T_1 \leq T_2$ and $T_2 \leq T_3$ imply $T_1 \leq T_3$. Thus we have equivalence classes of theories under mutual interpretability, partially ordered by the interpretability relation.

Examples 5.7.4. First-order arithmetic is interpretable in second-order arithmetic, and second-order arithmetic is interpretable in set theory. More generally, for all $n \geq 1$, nth-order arithmetic is interpretable in $(n+1)$st-order arithmetic and in set theory. It can be shown that $(n+1)$st-order arithmetic and set theory are not interpretable in nth order arithmetic. In particular, second-order arithmetic is not interpretable in first-order arithmetic. Results of this kind follow from Gödel's Second Incompleteness Theorem. We have

$$\mathsf{Z}_1 < \mathsf{Z}_2 < \cdots < \mathsf{Z}_n < \mathsf{Z}_{n+1} < \cdots < \mathsf{Z}_\infty < \mathsf{ZC} < \mathsf{ZFC}$$

where $T_1 < T_2$ means that T_1 is "weaker than" T_2, i.e., T_2 is "stronger than" T_1, i.e., T_1 is interpretable in T_2 and not vice versa.

Remark 5.7.5 (the Gödel hierarchy)**.** The partial ordering of foundational theories under interpretability is sometimes known as *the Gödel hierarchy.* This hierarchy is of obvious foundational interest.

Remark 5.7.6. The foundational significance of interpretability is highlighted by the following observations. If T_1 is interpretable in T_2, then:

1. Consistency of T_2 implies consistency of T_1.

2. Essential incompleteness of T_1 implies essential incompleteness of T_2.

3. Effective essential incompleteness of T_1 implies effective essential incompleteness of T_2.

5.8 Beth's Definability Theorem

In this section we consider implicit definitional extensions of theories. We state and prove Beth's Definability Theorem, which says that an implicit definitional extension of a theory T is always equivalent to an explicit definitional extension of the same theory, T.

We consider only the case of predicates, but operations can be handled similarly. Let $T = (L, S)$ be a theory, and let P be an n-ary predicate of L.

Definition 5.8.1.

1. We say that T *explicitly defines* P if there exists an L-formula D not involving P with free variables $x_1, \ldots, x_n$ such that
$$T \vdash \forall x_1 \cdots \forall x_n \, (Px_1 \cdots x_n \Leftrightarrow D).$$

2. We say that T *implicitly defines* P if, letting P' be a new n-ary predicate and letting $T' = T[P/P']$, we have
$$T \cup T' \vdash \forall x_1 \cdots \forall x_n \, (Px_1 \cdots x_n \Leftrightarrow P'x_1 \cdots x_n).$$

Theorem 5.8.2 (Beth's Definability Theorem)**.** T explicitly defines P if and only if T implicitly defines P.

Proof. Assume first that T explicitly defines P, say
$$T \vdash \forall x_1 \cdots \forall x_n \, (Px_1 \cdots x_n \Leftrightarrow D).$$

It follows that
$$T' \vdash \forall x_1 \cdots \forall x_n \, (P'x_1 \cdots x_n \Leftrightarrow D).$$

Hence
$$T \cup T' \vdash \forall x_1 \cdots \forall x_n \, (Px_1 \cdots x_n \Leftrightarrow P'x_1 \cdots x_n),$$
i.e., T implicitly defines P.

Conversely, assume that T implicitly defines P, i.e.,
$$T \cup T' \vdash \forall x_1 \cdots \forall x_n \, (Px_1 \cdots x_n \Leftrightarrow P'x_1 \cdots x_n).$$

By the Compactness Theorem, there are finitely many axioms $A_1, \ldots, A_k \in S$ such that
$$(A \wedge A') \Rightarrow \forall x_1 \cdots \forall x_n \, (Px_1 \cdots x_n \Leftrightarrow P'x_1 \cdots x_n)$$

is logically valid, where $A = A_1 \wedge \cdots \wedge A_k$ and $A' = A[P/P']$. Introducing parameters $a_1, \ldots, a_n$, we see that

$$(A \wedge A') \Rightarrow (Pa_1 \cdots a_n \Leftrightarrow P'a_1 \cdots a_n) \tag{5.1}$$

is logically valid. It follows quasitautologically that

$$(A \wedge Pa_1 \cdots a_n) \Rightarrow (A' \Rightarrow P'a_1 \cdots a_n) \tag{5.2}$$

is logically valid. By the Interpolation Theorem 3.5.1, we can find an L-formula D with free variables $x_1, \ldots, x_n$ such that $D[x_1/a_1, \ldots, x_n/a_n]$ is an interpolant for (5.2). Thus

$$(A \wedge Pa_1 \cdots a_n) \Rightarrow D[x_1/a_1, \ldots, x_n/a_n] \tag{5.3}$$

and

$$D[x_1/a_1, \ldots, x_n/a_n] \Rightarrow (A' \Rightarrow P'a_1 \cdots a_n) \tag{5.4}$$

are logically valid, and P and P' do not occur in D. Since (5.4) is logically valid, it follows that

$$D[x_1/a_1, \ldots, x_n/a_n] \Rightarrow (A \Rightarrow Pa_1 \cdots a_n) \tag{5.5}$$

is logically valid. From the logical validity of (5.3) and (5.5), it follows quasitautologically that

$$A \Rightarrow (Pa_1 \cdots a_n \Leftrightarrow D[x_1/a_1, \ldots, x_n/a_n]) \tag{5.6}$$

is logically valid. Hence

$$A \Rightarrow \forall x_1 \cdots \forall x_n \, (Px_1 \cdots x_n \Leftrightarrow D)$$

is logically valid, so

$$T \vdash \forall x_1 \cdots \forall x_n \, (Px_1 \cdots x_n \Leftrightarrow D).$$

Thus we see that T explicitly defines P. $\square$

Chapter 6

Arithmetization of Predicate Calculus

6.1 Primitive Recursive Arithmetic

Definition 6.1.1. To each natural number n we associate a variable-free PRA-term $\underline{n}$ as follows: $\underline{0} = 0$, $\underline{n+1} = \underline{S}\,\underline{n}$. Thus

$$\underline{n} = \underbrace{\underline{S}\cdots\underline{S}}_{n}\,0.$$

These terms are known as *numerals*.

Theorem 6.1.2. Let f be a k-ary primitive recursive function. Then for all k-tuples of natural numbers $m_1, \ldots, m_k$ we have

$$\mathsf{PRA} \;\vdash\; \underline{f}\;\underline{m_1}\cdots\underline{m_k} \;=\; \underline{f(m_1,\ldots,m_k)}\;.$$

Proof. □

6.2 Interpretability of PRA in Z_1

6.3 Gödel Numbers

Let L be a countable language. Assume that to all the sorts σ, predicates P, and operations f of L have been assigned distinct positive integers $\#(\sigma)$, $\#(P)$, $\#(f)$ respectively. As usual, let V be the set of parameters.

Definition 6.3.1 (Gödel numbers)**.** To each L-V-term t and L-V-formula A we assign a positive integer, the *Gödel number* of t or of A, denoted $\#(t)$ or

$\#(A)$, respectively.

$$
\begin{array}{lcll}
\#(v_i^\sigma) & = & 2 \cdot 3^{\#(\sigma)} \cdot 5^i & \\
\#(a_i^\sigma) & = & 2^2 \cdot 3^{\#(\sigma)} \cdot 5^i & \\
\#(f t_1 \cdots t_n) & = & 2^3 \cdot 3^{\#(f)} \cdot p_2^{\#(t_1)} \cdots p_{n+1}^{\#(t_n)} & \text{if } f \text{ is an } n\text{-ary operation} \\
\#(P t_1 \cdots t_n) & = & 2^4 \cdot 3^{\#(P)} \cdot p_2^{\#(t_1)} \cdots p_{n+1}^{\#(t_n)} & \text{if } P \text{ is an } n\text{-ary predicate} \\
\#(\neg A) & = & 2^5 \cdot 3^{\#(A)} & \\
\#(A \wedge B) & = & 2^6 \cdot 3^{\#(A)} \cdot 5^{\#(B)} & \\
\#(A \vee B) & = & 2^7 \cdot 3^{\#(A)} \cdot 5^{\#(B)} & \\
\#(A \Rightarrow B) & = & 2^8 \cdot 3^{\#(A)} \cdot 5^{\#(B)} & \\
\#(A \Leftrightarrow B) & = & 2^9 \cdot 3^{\#(A)} \cdot 5^{\#(B)} & \\
\#(\forall v\, A) & = & 2^{10} \cdot 3^{\#(v)} \cdot 5^{\#(A)} & \\
\#(\exists v\, A) & = & 2^{11} \cdot 3^{\#(v)} \cdot 5^{\#(A)} &
\end{array}
$$

Definition 6.3.2. The language L is said to be *primitive recursive* if the following items are primitive recursive.

$$
\begin{array}{lcll}
\mathrm{Sort}(x) & \equiv & x = \#(\sigma) & \text{for some sort } \sigma \\
\mathrm{Pred}(x) & \equiv & x = \#(P) & \text{for some predicate } P \\
\mathrm{Op}(x) & \equiv & x = \#(f) & \text{for some operation } f \\
\mathrm{arity}(\#(P)) & = & n & \text{if } P \text{ is an } n\text{-ary predicate} \\
\mathrm{arity}(\#(f)) & = & n & \text{if } f \text{ is an } n\text{-ary operation} \\
\mathrm{sort}(\#(P), i) & = & \#(\sigma_i) & \text{if } 1 \leq i \leq n \text{ and } P \text{ is an } n\text{-ary predicate of} \\
 & & & \text{type } (\sigma_1, \ldots, \sigma_n) \\
\mathrm{sort}(\#(f), i) & = & \#(\sigma_i) & \text{if } 1 \leq i \leq n \text{ and } \mathrm{sort}(\#(f)) = \#(\tau) \text{ and } f \text{ is an } n\text{-ary operation} \\
 & & & \text{of type } (\sigma_1, \ldots, \sigma_n, \tau).
\end{array}
$$

Lemma 6.3.3. If L is primitive recursive, then the following are primitive recursive.

$$
\begin{array}{lcll}
\mathrm{Var}(x) & \equiv & x = \#(v) & \text{for some variable } v \\
\mathrm{Param}(x) & \equiv & x = \#(a) & \text{for some parameter } a \\
\mathrm{Term}(x) & \equiv & x = \#(t) & \text{for some term } t \\
\mathrm{ClTerm}(x) & \equiv & x = \#(t) & \text{for some closed term } t \\
\mathrm{AtFml}(x) & \equiv & x = \#(A) & \text{for some atomic formula } A \\
\mathrm{Fml}(x) & \equiv & x = \#(A) & \text{for some formula } A \\
\mathrm{sort}(\#(t)) & = & \#(\sigma) & \text{if } t \text{ is a term of sort } \sigma
\end{array}
$$

Proof. We have

$$\mathrm{Var}(x) \equiv (x)_0 = 1 \wedge x = 2^{(x)_0} \cdot 3^{(x)_1} \cdot 5^{(x)_2} \wedge \mathrm{Sort}((x)_1)$$

and

$$\mathrm{Param}(x) \quad \equiv \quad (x)_0 = 2 \wedge x = 2^{(x)_0} \cdot 3^{(x)_1} \cdot 5^{(x)_2} \wedge \mathrm{Sort}((x)_1) \ .$$

To show that the predicate $\mathrm{Term}(x)$ is primitive recursive, we first show that the function $\mathrm{sort}(x)$ is primitive recursive, where $\mathrm{sort}(\#(t)) = \#(\sigma)$ if t is a term of sort σ, $\mathrm{sort}(x) = 0$ otherwise. Put $\mathrm{lh}(x) =$ least $w < x$ such that $(x)_w = 0$. We then have

$$\mathrm{sort}(x) \quad = \quad \begin{cases} (x)_1 & \text{if } \mathrm{Var}(x) \vee \mathrm{Param}(x)\,, \\ \mathrm{sort}((x)_1, \mathrm{lh}(x) \dot{-} 1) & \text{if } (x)_0 = 3 \wedge \mathrm{Op}((x)_1) \wedge (+)\,, \\ 0 & \text{otherwise}\,, \end{cases}$$

where

$$\begin{aligned} (+) \qquad & \mathrm{arity}((x)_1) = \mathrm{lh}(x) \dot{-} 2 \ \wedge \ x = \textstyle\prod_{i=0}^{\mathrm{lh}(x) \dot{-} 1} p_i^{(x)_i} \\ & \wedge \ (\forall i < \mathrm{lh}(x) \dot{-} 2) \ (\mathrm{sort}((x)_{i+2}) = \mathrm{sort}((x)_1, i+1)) \ . \end{aligned}$$

Then

$$\mathrm{Term}(x) \quad \equiv \quad \mathrm{sort}(x) > 0 \ .$$

For closed terms, define $\mathrm{clsort}(x)$ like $\mathrm{sort}(x)$ replacing $\mathrm{Var}(x) \vee \mathrm{Param}(x)$ by $\mathrm{Param}(x)$. We then have

$$\mathrm{ClTerm}(x) \quad \equiv \quad \mathrm{clsort}(x) > 0 \ .$$

For formulas we have

$$\mathrm{AtFml}(x) \quad \equiv \quad ((x)_0 = 4 \wedge \mathrm{Pred}((x)_1) \wedge (+))$$

and

$$\begin{aligned} \mathrm{Fml}(x) \ \equiv \ & \mathrm{AtFml}(x) \vee ((x)_0 = 5 \wedge x = 2^{(x)_0} \cdot 3^{(x)_1} \wedge \mathrm{Fml}((x)_1)) \\ & \vee (6 \le (x)_0 \le 9 \wedge x = 2^{(x)_0} \cdot 3^{(x)_1} \cdot 5^{(x)_2} \wedge \mathrm{Fml}((x)_1) \wedge \mathrm{Fml}((x)_2)) \\ & \vee (10 \le (x)_0 \le 11 \wedge x = 2^{(x)_0} \cdot 3^{(x)_1} \cdot 5^{(x)_2} \wedge \mathrm{Var}((x)_1) \wedge \mathrm{Fml}((x)_2)) \end{aligned}$$

and this completes the proof. □

Lemma 6.3.4 (substitution)**.** There is a primitive recursive function $\mathrm{sub}(x, y, z)$ such that for any formula A and any variable v and any closed term t,

$$\mathrm{sub}(\#(A), \#(v), \#(t)) \quad = \quad \#(A[v/t]) \ .$$

Proof.

$$\mathrm{sub}(x, y, z) \ = \ \begin{cases} z & \text{if } x = y, \\ 2^{(x)_0} \cdot 3^{(x)_1} \cdot \prod_{i=2}^{\mathrm{lh}(x) \dot{-} 1} p_i^{\mathrm{sub}((x)_i, y, z)} & \text{if } 3 \le (x)_0 \le 4, \\ 2^{(x)_0} \cdot 3^{\mathrm{sub}((x)_1, y, z)} & \text{if } (x)_0 = 5, \\ 2^{(x)_0} \cdot 3^{\mathrm{sub}((x)_1, y, z)} \cdot 5^{\mathrm{sub}((x)_2, y, z)} & \text{if } 6 \le (x)_0 \le 9, \\ 2^{(x)_0} \cdot 3^{(x)_1} \cdot 5^{\mathrm{sub}((x)_2, y, z)} & \text{if } 10 \le (x)_0 \le 11 \wedge (x)_1 \ne y, \\ x & \text{otherwise.} \end{cases}$$

$\square$

Lemma 6.3.5. If L is primitive recursive, then the predicate

$$\mathrm{Snt}(x) \quad \equiv \quad x = \#(A) \text{ for some sentence } A$$

is primitive recursive.

Proof. Recall that, by Exercise 2.1.10, a formula A is a sentence if and only if $A[v/a] = A$ for all variables v occurring in A. Note also that if $y = \#(v_i^\sigma)$ then $2y = \#(a_i^\sigma)$. Thus we have

$$\mathrm{Snt}(x) \;\equiv\; \mathrm{Fml}(x) \wedge (\forall y < x)\,(\mathrm{Var}(y) \Rightarrow x = \mathrm{sub}(x, y, 2y)) \ .$$

$\square$

Lemma 6.3.6. There is a primitive recursive function $\mathrm{num}(x)$ such that

$$\mathrm{num}(n) \quad = \quad \#(\underline{n})$$

for any nonnegative integer n.

Proof. The recursion equations for $\mathrm{num}(x)$ are

$$\begin{aligned} \mathrm{num}(0) \quad &= \quad \#(\underline{0}) \ , \\ \mathrm{num}(x+1) \quad &= \quad 2^3 \cdot 3^{\#(\underline{S})} \cdot 5^{\mathrm{num}(x)} \ . \end{aligned}$$

$\square$

6.4 Undefinability of Truth

In this section, let T be a theory which includes PRA. For example, we could take T to be PRA itself. Or, by Section 6.2, we could take T to be an appropriate definitional extension of Z_1 or Z_2 or ZFC.

Lemma 6.4.1 (Self-Reference Lemma)**.** Let L be the language of T. Let A be an L-formula with a free number variable x. Then we can find an L-formula B such that

$$T \quad \vdash \quad B \Leftrightarrow A[x/\underline{\#(B)}\,] \ .$$

The free variables of B are those of A except for x. In particular, if x is the only free variable of A, then B is an L-sentence.

Proof. Put $d(z) = \mathrm{sub}(z, \#(x), \mathrm{num}(z))$. Thus d is a 1-ary primitive recursive function such that, if A is any L-formula containing the number variable x as a free variable, then $d(\#(A)) = \#(A[x/\underline{\#(A)}\,]$. Now given A as in the hypothesis of the lemma, let D be the formula $A[x/\underline{d}\,x]$, and let B be the formula $A[x/\underline{d}\,\underline{\#(D)}\,]$, i.e., $D[x/\underline{\#(D)}\,]$. Note that $d(\#(D)) = \#(B)$. It follows by Theorem 6.1.2 that $\mathsf{PRA} \vdash \underline{d}\,\underline{\#(D)} = \underline{\#(B)}$. Since T includes PRA, it follows that $T \vdash \underline{d}\,\underline{\#(D)} = \underline{\#(B)}$. Hence $T \vdash A[x/\underline{d}\,\underline{\#(D)}] \Leftrightarrow A[x/\underline{\#(B)}\,]$. In other words, $T \vdash B \Leftrightarrow A[x/\underline{\#(B)}\,]$. This completes the proof. $\square$

Definition 6.4.2. If M is any model of T, let True_M be the set of Gödel numbers of sentences that are true in M, i.e.,

$$\mathrm{True}_M \quad = \quad \{\#(B) \mid B \text{ is a sentence and } M \models B\}.$$

Definition 6.4.3. An *ω-model of* T is a model M of T such that the number domain of M is $\omega = \{0, 1, 2, \ldots\}$ and $0_M = 0$ and $S_M(n) = n + 1$ for all $n \in \omega$. More generally, if M is any model of T, we may assume that ω is identified with a subset of the number domain of M in such a way that $0_M = 0$ and $S_M(n) = n + 1$ for each $n \in \omega$. Thus each $n \in \omega$ is identified with the element of M that is denoted by $\underline{n}$, i.e., $n = v_M(\underline{n})$.

Theorem 6.4.4 (undefinability of truth)**.** If M is an ω-model of T, then True_M is not explicitly definable over M. More generally, if M is any model of T, then the characteristic function of True_M is is not included in the characteristic function of any subset of M that is explicitly definable over M.

Proof. Let X be a subset of the number domain of M which is explicitly definable over M. Let A be an L-formula with a free number variable x and no other free variables, such that A explicitly defines X over M. Applying Lemma 6.4.1 to the negation of A, we obtain an L-sentence B such that $T \vdash B \Leftrightarrow \neg A[x/\overline{\#(B)}]$. Since M is a model of T, it follows that $\#(B) \in \mathrm{True}_M$ if and only if $\#(B) \notin X$. Hence the characteristic function of True_M is not included in the characteristic function of X, q.e.d. $\square$

Corollary 6.4.5. Let $M = (\omega, +, \cdot, 0, 1, =)$ be the standard model of first-order arithmetic, Z_1. Then True_M is not explicitly definable over M. This[1] may be paraphrased by saying that arithmetical truth is not arithmetically definable.

Remark 6.4.6. With $M = (\omega, +, \cdot, 0, 1, =)$ as above, it can be shown that True_M is implicitly definable over M. (See also Exercise 6.4.7.) Thus the Beth's Definability Theorem does not hold for definability over this particular model.

Exercise 6.4.7. Let M be an ω-model of Z_1 or Z_2 or ZFC. Let Sat_M be the satisfaction relation on M. Show that Sat_M is implicitly definable over M.

6.5 The Provability Predicate

In this section, let L be a primitive recursive language, and let T be an L-theory which is primitive recursively axiomatizable. For example, T could be PRA itself, or T could be any of the mathematical or foundational theories discussed in Sections 5.2, 5.5, 5.6.

Definition 6.5.1. Choose a primitive recursive predicate Ax_T for the set of Gödel numbers of axioms of T. In terms of Ax_T show that various predicates

[1] This result is due to Tarski [5].

associated with T are primitive recursive. Introduce the provability predicate Pvbl_T by definition:

$$\text{Pvbl}_T(x) \;\Leftrightarrow\; \exists y\, \text{Prf}_T(x, y) \;.$$

Note that, for all L-sentences B, $\text{Pvbl}_T(\underline{\#(B)})$ is true if and only if $T \vdash B$.

Lemma 6.5.2 (derivability condition 1)**.** For any L-sentence A, if $T \vdash A$ then

$$\mathsf{PRA} \;\vdash\; \text{Pvbl}_T(\underline{\#(A)}) \;.$$

Proof. Suppose $T \vdash A$. Let p be a proof of A in T. Then $\text{Prf}_T(\#(A), \#(p))$ holds. Since $\text{Prf}_T(x, y)$ is a primitive recursive predicate, it follows by Theorem 6.1.2 that $\mathsf{PRA} \vdash \text{Prf}_T(\underline{\#(A)}, \underline{\#(p)})$. Hence $\mathsf{PRA} \vdash \text{Pvbl}_T(\underline{\#(A)})$, q.e.d. $\square$

Lemma 6.5.3 (derivability condition 2)**.** For any L-sentence A, we have

$$\mathsf{PRA} \;\vdash\; \text{Pvbl}_T(\underline{\#(A)}) \Rightarrow \text{Pvbl}_{\mathsf{PRA}}(\underline{\#(\text{Pvbl}_T(\underline{\#(A)}))}) \;.$$

Proof. This is just Lemma 6.5.2 formalized in PRA. The details of the formalization are in Section 6.7. $\square$

Lemma 6.5.4 (derivability condition 3)**.** For any L-sentences A and B, we have

$$\mathsf{PRA} \;\vdash\; \text{Pvbl}_T(\underline{\#(A \Rightarrow B)}) \Rightarrow (\text{Pvbl}_T(\underline{\#(A)}) \Rightarrow \text{Pvbl}_T(\underline{\#(B)})) \;.$$

Proof. This is a straightforward consequence of the fact that our rules of inference include modus ponens. $\square$

6.6 The Incompleteness Theorems

In this section, let T be a theory which is primitive recursively axiomatizable and includes PRA. For example, T could be PRA itself, or it could be an appropriate definitional extension of Z_1 or Z_2 or ZFC. As in Section 6.5, let Pvbl_T be a provability predicate for T.

Lemma 6.6.1. Let $A(x)$ be a PRA-formula with one free variable x. Then we can find a PRA-sentence B such that $\mathsf{PRA} \vdash B \Leftrightarrow A(\underline{\#(B)})$.

Proof. This is the Self-Reference Lemma 6.4.1 specialized to PRA. $\square$

Lemma 6.6.2. We can find a PRA-sentence S such that

$$\mathsf{PRA} \;\vdash\; S \Leftrightarrow \neg\, \text{Pvbl}_T(\underline{\#(S)}) \;. \tag{6.1}$$

Note that S is self-referential and says "I am not provable in T."

Proof. This is an instance of Lemma 6.6.1 with $A(x) \equiv \neg\, \text{Pvbl}_T(x)$. $\square$

Lemma 6.6.3. Let S be as in Lemma 6.6.2. If T is consistent, then $T \nvdash S$.

Proof. Suppose for a contradiction that $T \vdash S$. By Lemma 6.5.2 we have $\mathsf{PRA} \vdash \mathrm{Pvbl}_T(\underline{\#(S)})$. Hence by (6.1) it follows that $\mathsf{PRA} \vdash \neg S$. Since T includes PRA, we have $T \vdash \neg S$. Thus T is inconsistent. $\square$

Theorem 6.6.4 (the First Incompleteness Theorem)**.** If T is consistent, then we can find a sentence S' in the language of first-order arithmetic such that S' is true yet S' is not a theorem of T.

Proof. Let S be a PRA-sentence as in Lemma 6.6.2. By Lemma 6.6.3, $T \nvdash S$. It follows by (6.1) that S is true. As in Section 6.2, let S' be the translation of S into the language of first-order arithmetic. Thus S' is also true. By the results of Section 6.2, $\mathsf{PRA} \vdash S \Leftrightarrow S'$. Hence $T \vdash S \Leftrightarrow S'$. Hence $T \nvdash S'$. $\square$

Assume now that the primitive recursive predicate Ax_T has been chosen in such a way that $\mathsf{PRA} \vdash \forall x\,(\mathrm{Ax}_{\mathsf{PRA}}(x) \Rightarrow \mathrm{Ax}_T(x))$. It follows that $\mathsf{PRA} \vdash \forall x\,(\mathrm{Pvbl}_{\mathsf{PRA}}(x) \Rightarrow \mathrm{Pvbl}_T(x))$. In particular we have:

Lemma 6.6.5. For all PRA-sentences A, we have

$$\mathsf{PRA} \;\vdash\; \mathrm{Pvbl}_{\mathsf{PRA}}(\underline{\#(A)}) \Rightarrow \mathrm{Pvbl}_T(\underline{\#(A)})\ .$$

Definition 6.6.6. Con_T is defined to be the sentence $\neg\,\mathrm{Pvbl}_T(\underline{\#(0 \neq 0)})$. Note that Con_T is a PRA-sentence which asserts the consistency of T.

Theorem 6.6.7 (the Second Incompleteness Theorem)**.** If T is consistent, then $T \nvdash \mathrm{Con}_T$.

Proof. Let S be as in Lemma 6.6.2. By Theorem 6.6.4 we have $T \nvdash S$. Therefore, to show $T \nvdash \mathrm{Con}_T$, it suffices to show $T \vdash \mathrm{Con}_T \Rightarrow S$. Since T includes PRA, it suffices to show $\mathsf{PRA} \vdash \mathrm{Con}_T \Rightarrow S$. By (6.1) it suffices to show

$$\mathsf{PRA} \;\vdash\; \mathrm{Con}_T \Rightarrow \neg\,\mathrm{Pvbl}_T(\underline{\#(S)})\ . \tag{6.2}$$

But this is just Lemma 6.6.3 formalized in PRA.

Details: We need to prove (6.2). Reasoning in PRA, suppose $\mathrm{Pvbl}_T(\underline{\#(S)})$. By Lemma 6.5.3 we have $\mathrm{Pvbl}_{\mathsf{PRA}}(\underline{\#(\mathrm{Pvbl}_T(\underline{\#(S)}))})$. Moreover, from (6.1) and Lemma 6.5.2 we have $\mathrm{Pvbl}_{\mathsf{PRA}}(\underline{\#(S \Leftrightarrow \neg\,\mathrm{Pvbl}_T(\underline{\#(S)}))})$. Hence by Lemmas 6.5.2 and 6.5.4 we have $\mathrm{Pvbl}_{\mathsf{PRA}}(\underline{\#(\neg S)})$. By Lemma 6.6.5 it follows that $\mathrm{Pvbl}_T(\underline{\#(\neg S)})$. Hence by Lemmas 6.5.2 and 6.5.4 we have $\mathrm{Pvbl}_T(\underline{\#(0 \neq 0)})$, i.e., $\neg\,\mathrm{Con}_T$. This completes the proof. $\square$

Exercise 6.6.8. Show that $\mathsf{PRA} \vdash S \Leftrightarrow \mathrm{Con}_T$.

Exercise 6.6.9 (Rosser's Theorem)**.** Show that if T is as in Theorems 6.6.4 and 6.6.7, then T is incomplete.

Hint: Use the Self-Reference Lemma 6.4.1 to obtain a sentence B such that $\mathsf{PRA} \vdash B \Leftrightarrow A[x/\underline{\#(B)}]$, where $A[x/\underline{\#(B)}]$ says that for any proof p of B in T there exists a proof q of $\neg B$ in T such that $\#(q) < \#(p)$. Using the assumption that T is consistent, show that $T \nvdash B$ and $T \nvdash \neg B$.

Exercise 6.6.10. Give an example of a T as in Theorems 6.6.4 and 6.6.7 such that $T \vdash \neg\,\mathrm{Con}_T$.

Solution. We may take $T = \mathsf{PRA} + \neg\,\mathrm{Con}_{\mathsf{PRA}}$, or $T = \mathsf{Z}_1 + \neg\,\mathrm{Con}_{\mathsf{Z}_1}$, etc.

6.7 Proof of Lemma 6.5.3

FIXME Write this section!

Bibliography

[1] Neil D. Jones and Alan L. Selman. Turing machines and the spectra of first-order formulas. *Journal of Symbolic Logic*, 39:139–150, 1974.

[2] David Marker. *Model Theory: An Introduction*. Springer, 2002. VIII + 342 pages.

[3] Stephen G. Simpson. *Subsystems of Second Order Arithmetic*. Perspectives in Mathematical Logic. Springer-Verlag, 1999. XIV + 445 pages.

[4] Raymond M. Smullyan. *First-Order Logic*. Dover Publications, New York, 1995. XII + 158 pages.

[5] Alfred Tarski. *Introduction to Logic and to the Methodology of Deductive Sciences*. Oxford University Press, 4th edition, 1994. XXII + 229 pages.

Index