Gödel's Incompleteness Theorems

J. van Oosten
Department of Mathematics
Utrecht University

2009–2010

Contents

Chapter 1

Languages and Structures

1.1 Structure and Languages for the Real Numbers: an Example

This section is meant to serve as introduction and motivation for the formal definition of an abstract *language* in the next section.

When we say that the real numbers $\mathbb{R}$ form a *commutative ring with* 1, we mean that there are two *distinguished* elements 0 and 1, as well as *operations* + and $\cdot$, such that certain *axioms* hold, for example:

$$x{\cdot}(y+z) = (x{\cdot}y) + (x{\cdot}z)$$

This is to be read as: whenever real numbers are substituted for the *variables* x, y and z, we get an equality as above.

We call the whole of $\{0, 1, +, \cdot\}$ the *ring structure* of $\mathbb{R}$. Now of course you know there are plenty of other rings. For example, let X be any set. The power set $\mathcal{P}(X)$ can be made into a commutative ring with 1: take X for 1, $\emptyset$ for 0, and let for $U, V \subseteq X$, $U + V = (U \cup V) - (U \cap V)$ and $U{\cdot}V = U \cap V$.

Exercise 1 Check that this indeed gives a ring structure on $\mathcal{P}(X)$.

The example of $\mathcal{P}(X)$ makes it clear that the operation + does not, a priori, mean addition of numbers, but is an abstract symbol generally used for the operation in abelian groups; we might as well have used something like $a(x, y)$ and $m(x, y)$ instead of $x + y$ and $x{\cdot}y$, respectively, and written the distributivity axiom as

$$m(x, a(y, z)) = a(m(x, y), m(x, z))$$

Similarly, one should regard 0 and 1 as abstract symbols that only acquire meaning once they are *interpreted* in a particular set.

Many axioms for rings have a very simple form: they are equalities between *terms*, where a term is an expression built up using variables, the symbols 0 and 1, and the operation symbols $+$, $\cdot$ (and brackets). From simple equalities we can form more involved statements using *logical operations*: the operations $\wedge$ ("and"), $\vee$ ("or"), $\rightarrow$ ("if…then"), $\leftrightarrow$ ("if and only if") and $\neg$ ("not"); and *quantifiers* $\exists$ ("there is") and $\forall$ ("for all"). For example, if we want to express that $\mathbb{R}$ is in fact a field, we may write

$$\forall x(\neg(x = 0) \rightarrow \exists y(x{\cdot}y = 1))$$

or equivalently

$$\forall x \exists y(x = 0 \vee x{\cdot}y = 1)$$

We say that the statement $\forall x \exists y(x = 0 \vee x{\cdot}y = 1)$ "is true in $\mathbb{R}$" (of course, what we really mean is: in $\mathbb{R}$ together with the meaning of $0, 1, +, \cdot$). Such statements can be used to distinguish between various rings: for example, the statement

$$\exists x(x{\cdot}x = 1 + 1)$$

is true in $\mathbb{R}$ but not in the ring $\mathbb{Q}$, and the statement

$$\forall x(x{\cdot}x = x)$$

is true in the ring $\mathcal{P}(X)$ but not in $\mathbb{R}$.

Apart from operations on a set, one may also consider certain *relations*. In $\mathbb{R}$ we have the relation of *order*, expressed by $x < y$. As before, we might have used a different symbol for this relation, for example $L(x, y)$ ("x is less than y"). And we can form statements using this new symbol together with the old ones, for example

$$\forall x \forall y \forall z(x < y \rightarrow x + z < y + z)$$

which is one of the axioms for an *ordered ring*. In $\mathbb{R}$, the order relation is *definable* from the ring structure, because the statement

$$\forall x \forall y(x < y \leftrightarrow \exists z(\neg(z = 0) \wedge x + z{\cdot}z = y))$$

is true in $\mathbb{R}$. However, this statement is not true in the ordered ring $\mathbb{Q}$. Also the ring $\mathcal{P}(X)$ is (partially) ordered by $U \subset V$; in this ring, the order is definable, but now in a different way:

$$\forall x \forall y(x < y \leftrightarrow (\neg(x = y) \wedge x{\cdot}y = x))$$

In yet another way, the order in $\mathbb{Q}$ is definable from the ring structure. In this case, we use the theorem (first proved by Lagrange) which says that every natural number may be written as the sum of four squares. Since every positive rational number is the quotient of two positive natural numbers, we have:

$$x > 0 \leftrightarrow x \neq 0 \wedge \exists y_1 \cdots y_8 \, (x \cdot (y_1^2 + \cdots + y_4^2) = y_5^2 + \cdots + y_8^2)$$

for $x \in \mathbb{Q}$. Since $x < y$ is equivalent to $\exists z(z > 0 \wedge x + z = y)$, we can define the order on $\mathbb{Q}$ in terms of 0, + and $\cdot$ only.

We see that in general, when we wish to discuss a certain type of mathematical structures, we choose symbols for the distinguished elements, the operations and the relations which make up the structure, and using these we write down statements. The use of such statements is varied: they may be axioms, required to be true in all structures we wish to consider; they may be true in some, but not in others; or they may be used to *define* elements or subsets of a structure.

In Mathematical Logic, we study these statements, and their relation to mathematical structures, formally; in order to do this, *we define formal statements as mathematical objects*. This is done in the next section.

We shall see many examples of different types of structures in the coming sections.

1.2 Languages of First Order Logic

This section is purely "linguistic" and introduces the formal languages for first-order logic – or "predicate logic".

Definition 1.1 A *language* L is given by three sets of symbols: *constants*, *function symbols* and *relation symbols*. We may write

$$L = (\mathrm{con}(L), \mathrm{fun}(L), \mathrm{rel}(L))$$

Moreover, for each function symbol f and each relation symbol R the number n of arguments is specified, and called the *arity* of f (or R). If f or R has arity n, we say that it is an *n-ary* (or *n-place*) function (relation) symbol.

For example, the language of rings has two constants, 0 and 1, and two 2-place function symbols for addition and multiplication. There are no relation symbols.

The language of orders has one 2-place relation symbol (S or $<$) for "less than".
Given such a language L, one can build *terms* (to denote elements) and *formulas* (to state properties), using the following auxiliary symbols:

- An infinite set of *variables*. This set is usually left unspecified, and its elements are denoted by $x, y, z, \ldots$ or $x_0, x_1, \ldots$
- The equality symbol $=$
- The symbol $\perp$ ("absurdity")
- Connectives: the symbols $\wedge$ ("and") for *conjunction*, $\vee$ ("or") for *disjunction*, $\rightarrow$ ("if. . . then") for *implication* and $\neg$ ("not") for *negation*
- Quantifiers: the *universal quantifier* $\forall$ ("for all") and the *existential quantifier* $\exists$ ("there exists")
- Some readability symbols, like the comma, and brackets.

Definition 1.2 The set of *terms* of a language L is inductively defined as follows:

- any constant c of L is a term of L;
- any variable x is a term of L;
- if $t_1, \ldots, t_n$ is an n-tuple of terms of L and f is an n-place function symbol of L, then $f(t_1, \ldots, t_n)$ is a term of L.

A term which does not contain variables (and hence is built up from constants and function symbols alone) is called *closed*.

Examples

a) Suppose L has a constant c and a 2-place function symbol f. The following are terms of L: $x, y, c, f(x, c), f(f(x, c), c), \ldots$

b) Suppose L has no function symbols. The only terms are variables and constants.

Definition 1.3 The set of *formulas* of a given language L is inductively defined as follows:

- If t and s are terms of L, then $(t = s)$ is a formula of L.

- If $t_1, \dots, t_n$ is an n-tuple of terms of L and R is an n-place relation symbol of L, then $R(t_1, \dots, t_n)$ is a formula of L.
- $\bot$ is a formula of L.
- If φ and ψ are formulas of L, then so are $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$ and $(\neg\varphi)$.
- If φ is a formula of L, and x is a variable, then also $\forall x\varphi$ and $\exists x\varphi$ are formulas of L.

Remarks/Examples.

a) Given a language L, let V be the set of variables, and A the set of auxiliary symbols that we have listed. Let $S = L \cup V \cup A$. Then formally, terms of L and formulas of L are finite tuples of elements of S.

b) However, the sets of terms of a language and of formulas of a language have a more meaningful structure. Suppose t is a term. Then there are three possibilities: t is a variable, t is a constant, or there is an n-place function symbol f of L, and terms $t_1, \dots, t_n$, such that $t = f(t_1, \dots, t_n)$. The terms $t_1, \dots, t_n$ have the property that each one of them contains *fewer* function symbols of L than t. One uses this to prove properties of terms "by induction on the number of function symbols occurring in them". Similarly, one can prove properties of formulas by induction on the number of symbols from the set $\{\wedge, \vee, \rightarrow, \neg, \forall, \exists\}$ in them. If this number is zero, we call the formula *atomic*.

c) The use of brackets and commas is *only* for the sake of readability and to avoid ambiguity, such as $\varphi \vee \psi \rightarrow \chi$. Outermost brackets are usually omitted.

d) Suppose the language L has one constant c, one 2-place function symbol f and one 3-place relation symbol R. Then

$$\forall x \forall y R(c, x, f(y, c))$$
$$\forall x(x = f(x, x) \rightarrow \exists y R(x, c, y))$$
$$R(f(x, f(c, f(y, c))), c, y) \wedge (x = y \vee \neg R(c, c, x))$$

are formulas of L (note how we use the brackets!), but

$$\forall R \neg R(x, x, c)$$

isn't (this might be called a "second order formula"; quantifying over relations).

Free and bound variables. Roughly speaking, a variable which is "quantified away" in a formula, is called *bound* in that formula; otherwise, it is called *free*.

For example, in the formula

$$\forall x(R(x,y) \to \exists z P(x,z))$$

the variables x and z are bound whereas y is free. The x in "$\forall x$" is not considered to be either free or bound, nor z in "$\exists z$".

The intuition is, that the formula above states a property of the variable y but not of the variables x, z; it should mean the same thing as the formula

$$\forall u(R(u,y) \to \exists v P(u,v))$$

A formula with no free variables is called *closed*, or a *sentence*. Such a formula should be thought of as an *assertion*.

It is an unfortunate consequence of the way we defined formulas, that expressions like

$$\forall x \forall y \forall x R(x,y)$$
$$\forall y(R(x,y) \to \forall x R(x,x))$$

are formulas. The first one has the strange property that the variable x is bound twice; and the second one has the undesirable feature that the variable x occurs both bound and free. In practice, we shall always stick to the following

CONVENTION ON VARIABLES *In formulas, a variable will always be either bound, or free, but not both; and if it is bound, it is only bound once*

Definition 1.4 (Substitution) Suppose φ is a formula of L, and t a term of L. By the *substitution* $\varphi[t/x]$ we mean the formula which results by replacing each occurrence of the variable x by the term t, provided x is a free variable in φ, and no variable in the term t becomes bound in φ (in this definition, the Convention on variables is in force!).

Examples. Suppose φ is the formula $\forall x R(x,y)$. If t is the term $f(u,v)$, then $\varphi[t/x]$ is just φ, since x is bound in φ; $\varphi[t/y]$ is $\forall x R(x, f(u,v))$.

Suppose t is the term $f(x,y)$. Now the substitution $\varphi[t/y]$ presents us with a problem; if we carry out the replacement of y by t we get $\forall x R(x, f(x,y))$, which intuitively does not "mean" that the property expressed by φ, holds for the element denoted by t! Therefore, we say that the substitution is not defined in this case. In practice though, as said before, we shall consider φ as the "same" formula as $\forall uR(u,y)$, and now the substitution makes sense: we get $\forall uR(u,f(x,y))$.

If the term t is closed (in particular, if t is a constant), the substitution $\varphi[t/x]$ is always defined, as is easy to see.

First order logic and other kinds of logic. In these lecture notes, we shall limit ourselves to the study of "first order logic", which is the study of the formal languages and formulas as we have described here, and their relation to structures, as we will see in the next section.

This logic has good mathematical properties, but it has also severe limitations. Our variables denote, as we shall see, elements of structures. So we can only say things about *all elements* of a structure, not about all subsets, or about sequences of elements. For example, consider the language of orders: we have a 2-place relation symbol $<$ for "less than". We can express that $<$ really is a partial order:

$$(\forall x\neg(x<x))\wedge(\forall x\forall y\forall z((x<y\wedge y<z)\rightarrow x<z))$$

and that $<$ is a linear order:

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

but we can *not* express that $<$ is a well-order, since for that we have to say something about *all* subsets.

It is possible to consider logics where such statements can be done: these are called "higher order" logics. There are also logics in which it is possible to form the conjunction, or disjunction, of an infinite set of formulas (so, formulas will be infinite objects in such a logic).

1.3 Structures for first order logic

In this section we consider a fixed but arbitrary first order language L, and discuss what it means to have a *structure for* L.

Definition 1.5 An L-*structure* M consists of a nonempty set, also denoted M, together with the following data:

- for each constant c of L, an element c^M of M;

- for each n-place function symbol f of L, a function

$$f^M : M^n \rightarrow M$$

- for each n-place relation symbol R of L, a subset

$$R^M \subseteq M^n$$

We call the element c^M the *interpretation* of c in M, and similarly, f^M and R^M are called the interpretations of f and R, respectively.

Given an L-structure M, we consider the language L_M (the *language of the structure* M): L_M is L together with, for each element m of M, an extra constant (also denoted m). Here it is assumed that $\text{con}(L) \cap M = \emptyset$. If we stipulate that the interpretation in M of each new constant m is the element m, then M is also an L_M-structure.

Definition 1.6 (Interpretation of terms) For each closed term t of the language L_M, we define its interpretation t^M as element of M, by induction on t, as follows. If t is a constant, then its interpretation is already defined since M is an L_M-structure. If t is of the form $f(t_1, \ldots, t_n)$ then also $t_1, \ldots, t_n$ are closed terms of L_M, so by induction hypothesis their interpretations $t_1^M, \ldots, t_n^M$ have already been defined; we put

$$t^M = f^M(t_1^M, \ldots, t_n^M)$$

Next, we define for a closed formula φ of L_M what it means that "φ is *true in* M" (other ways of saying this, are: φ *holds* in M, or M *satisfies* φ). Notation:

$$M \models \varphi$$

Definition 1.7 (Interpretation of formulas) For a closed formula φ of L_M, the relation $M \models \varphi$ is defined by induction on φ:

- If φ is an atomic formula, it is equal to $\bot$, of the form $(t_1 = t_2)$, or of the form $R(t_1, \ldots, t_n)$ with $t_1, t_2, \ldots, t_n$ closed terms; define:

$$\begin{array}{rcl} M \models \bot & \textit{never} \text{ holds} & \\ M \models (t_1 = t_2) & \text{iff} & t_1^M = t_2^M \\ M \models R(t_1, \ldots, t_n) & \text{iff} & (t_1^M, \ldots, t_n^M) \in R^M \end{array}$$

  where the t_i^M are the interpretations of the terms according to definition 1.6, and R^M the interpretation of R in the structure M.

- If φ is of the form $(\varphi_1 \wedge \varphi_2)$ define

$$M \models \varphi \quad \text{iff} \quad M \models \varphi_1 \text{ and } M \models \varphi_2$$

- If φ is of the form $(\varphi_1 \vee \varphi_2)$ define

$$M \models \varphi \quad \text{iff} \quad M \models \varphi_1 \text{ or } M \models \varphi_2$$

(the "or" is to be read as *inclusive*: as either…or, or both)

- If φ is of the form $(\varphi_1 \to \varphi_2)$ define

$$M \models \varphi \quad \text{iff} \quad M \models \varphi_2 \text{ whenever } M \models \varphi_1$$

- If φ is of the form $(\neg\psi)$ define

$$M \models \varphi \quad \text{iff} \quad M \not\models \psi$$

(here $\not\models$ means "not $\models$")

- If φ is of the form $\forall x\psi$ define

$$M \models \varphi \quad \text{iff} \quad M \models \psi[m/x] \text{ for all } m \in M$$

- If φ is of the form $\exists x\psi$ define

$$M \models \varphi \quad \text{iff} \quad M \models \psi[m/x] \text{ for some } m \in M$$

(in the last two clauses, $\psi[m/x]$ results by substitution of the new constant m for x in ψ)

In a way, this truth definition 1.7 simply translates the formulas of L_M (and hence, of L) into ordinary language. For example, if R is a binary (2-place) relation symbol of L and M is an L-structure, then $M \models \forall x\exists y R(x, y)$ if and only if for each $m \in M$ there is an $n \in M$ such that $(m, n) \in R^M$; that is, R^M contains the graph of a function $M \to M$.

Validity and Equivalence of Formulas

The symbol $\leftrightarrow$ is usually treated as an abbreviation: $\varphi \leftrightarrow \psi$ abbreviates $(\varphi \to \psi) \wedge (\psi \to \varphi)$. So, $M \models \varphi \leftrightarrow \psi$ if and only if the two statements $M \models \varphi$ and $M \models \psi$ are either both true or both false. We call the formulas φ and ψ *(logically) equivalent* if this is the case for all M.

Note, that the closed formula $\exists x(x = x)$ is always true, in every structure (this is a formula of every language!), since structures are required to be nonempty. In general, if φ is a formula in a language L such that for every L-structure M and every substitution of constants from M for the free variables of φ, $M \models \varphi$, then φ is called *valid.* So, φ and ψ are equivalent formulas, if and only if the formula

$$\varphi \leftrightarrow \psi$$

is valid.

The next couple of exercises provide you with a number of useful equivalences between formulas.

Exercise 2 Show that the following formulas are valid:

$\varphi \leftrightarrow \neg\neg\varphi$
$\neg\varphi \leftrightarrow (\varphi \rightarrow \bot)$
$(\varphi \rightarrow \psi) \leftrightarrow (\neg\varphi \vee \psi)$
$(\varphi \vee \psi) \leftrightarrow \neg(\neg\varphi \wedge \neg\psi)$
$(\varphi \wedge \psi) \leftrightarrow \neg(\neg\varphi \vee \neg\psi)$
$\exists x\varphi \leftrightarrow \neg\forall x\neg\varphi$
$\forall x\varphi \leftrightarrow \neg\exists x\neg\varphi$

$(\varphi \wedge (\psi \vee \chi)) \leftrightarrow ((\varphi \wedge \psi) \vee (\varphi \wedge \chi))$
$(\varphi \vee (\psi \wedge \chi)) \leftrightarrow ((\varphi \vee \psi) \wedge (\varphi \vee \chi))$
$(\varphi \rightarrow (\psi \vee \chi)) \leftrightarrow ((\varphi \rightarrow \psi) \vee (\varphi \rightarrow \chi))$
$(\varphi \rightarrow (\psi \wedge \chi)) \leftrightarrow ((\varphi \rightarrow \psi) \wedge (\varphi \rightarrow \chi))$

In the following, assume that x does not occur in φ
$(\varphi \rightarrow \exists x\psi) \leftrightarrow \exists x(\varphi \rightarrow \psi)$
$(\exists x\psi \rightarrow \varphi) \leftrightarrow \forall x(\psi \rightarrow \varphi)$
$(\forall x\psi \rightarrow \varphi) \leftrightarrow \exists x(\psi \rightarrow \varphi)$

Exercise 3 Prove that for every formula φ, φ is equivalent to a formula which starts with a string of quantifiers, followed by a formula in which no quantifiers occur. Such a formula is called *in prenex normal form.*

Exercise 4 a) Let φ be a formula in which no quantifiers occur. Show that φ is logically equivalent to a formula of the form:

$$\psi_1 \vee \cdots \vee \psi_k$$

where each ψ_i is a conjunction of atomic formulas and negations of atomic formulas. This form is called a *disjunctive normal form* for φ.

b) Let φ as in a); show that φ is also equivalent to a formula of the form

$$\psi_1 \wedge \cdots \wedge \psi_k$$

where each ψ_i is a disjunction of atomic formulas and negations of atomic formulas. This form is called a *conjunctive normal form* for φ.

In the following exercises you are asked to give L-sentences which "express" certain properties of structures. This means: give an L-sentence ϕ such that for every L-structure M it holds that $M \models \phi$ if and only if the structure M has the given property.

Exercise 5 Let L be the empty language. An L-structure is "just" a nonempty set M.

Express by means of an L-sentence that M has exactly 4 elements.

Exercise 6 Let L be a language with one 2-place relation symbol R. Give L-sentences which express:

a) R is an equivalence relation.

b) There are exactly 2 equivalence classes.

[That is, e.g. for a): $M \models \phi$ if and only if R^M is an equivalence relation on M, etc.]

Exercise 7 Let L be a language with just one 1-place function symbol F. Give an L-sentence ϕ which expresses that F is a bijective function.

Exercise 8 Let L be the language with just the 2-place function symbol $\cdot$. We consider the L-structures $\mathbb{Z}$ and $\mathbb{Q}$ where $\cdot$ is interpreted as ordinary multiplication.

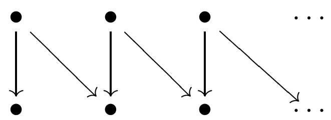
a) "Define" the numbers 0 and 1. That is, give L-formulas $\varphi_0(x)$ and $\varphi_1(x)$ with one free variable x, such that in both $\mathbb{Q}$ and $\mathbb{Z}$, $\varphi_i(a)$ is true exactly when $a = i$ $(i = 0, 1)$.

b) Give an L-sentence which is true in $\mathbb{Z}$ but not in $\mathbb{Q}$.

1.4 Examples of first order languages and structures

1.4.1 Graphs

A directed graph is a structure with vertices (points) and edges (arrows) between them, such as:



The language L_{graph} of directed graphs has two 1-place relation symbols, E and V (for "edge" and "vertex"), and two 2-place relation symbols S and T (for "source" and "target"; $S(x, y)$ will mean "the vertex x is the source of the edge y").

An L_{graph}-structure is a set G together with two subsets E^G, V^G of G, and two subsets S^G, T^G of G^2. G is a directed graph precisely when G satisfies the following 'axioms' for directed graphs:

$$\begin{array}{ll}
\forall x(E(x) \vee V(x)) & \forall x \neg(E(x) \wedge V(x)) \\
\forall x \forall y(S(x, y) \rightarrow (V(x) \wedge E(y))) & \forall x \forall y(T(x, y) \rightarrow (V(x) \wedge E(y))) \\
\forall x \forall y \forall z((S(x, z) \wedge S(y, z)) \rightarrow x = y) & \forall x \forall y \forall z((T(x, z) \wedge T(y, z)) \rightarrow x = y) \\
\forall z(E(z) \rightarrow \exists x \exists y(S(x, z) \wedge T(y, z))) &
\end{array}$$

1.4.2 Local Rings

The language L_{rings} of rings has constants 0 and 1, two 2-place function symbols for multiplication and addition, denoted $\cdot$ and $+$. There are no relation symbols.

A *commutative ring with* 1 is an L_{rings}-structure which satisfies the axioms for commutative rings with 1:

$$\begin{array}{ll}
\forall x(x + 0 = x) & \forall x(x \cdot 1 = x) \\
\forall xy(x + y = y + x) & \forall xy(x \cdot y = y \cdot x) \\
\forall xyz(x + (y + z) = (x + y) + z) & \forall xyz(x \cdot (y \cdot z) = (x \cdot y) \cdot z) \\
\forall x \exists y(x + y = 0) & \forall xyz(x \cdot (y + z) = x \cdot y + x \cdot z)
\end{array}$$

(We have started to abbreviate a string of quantifiers of the same kind: instead of $\forall x \forall y$ write $\forall xy$)

A *local ring* is a commutative ring with 1 which has exactly one maximal ideal. This is a condition that involves quantifying over subsets (ideals) of the ring, and cannot be formulated in first order logic. However, one can show that a commutative ring with 1 is local, precisely when for each pair of elements x, y it holds that if $x + y$ is a unit, then either x or y must be a unit. That is, a commutative ring R with 1 is local, if and only if

$$R \models \forall xy(\exists z(z \cdot (x + y) = 1) \rightarrow (\exists v(v \cdot x = 1) \vee \exists w(w \cdot y = 1)))$$

Exercise 9 Let L be L_{rings} together with an extra 1-place relation symbol I. Give L-formulas which express that the subset defined by I is:

a) an ideal;

b) a prime ideal;

c) a maximal ideal.

1.4.3 Vector Spaces

Fix a field k. We can write down a language L_k of first order logic, and axioms in this language, such that the L_k-structures which satisfy the axioms are precisely the k-vector spaces.

The language L_k has a constant 0 and a binary function symbol + to describe the abelian group structure. Furthermore, it has a 1-place function symbol f_m for every element m of k, to describe scalar multiplication. Apart from the axioms for an abelian group (which are the left side of the axioms for rings given above), there are the axioms:

$$\begin{array}{ll} f_m(0) = 0 & \forall xy(f_m(x+y) = f_m(x) + f_m(y)) \\ \forall x(f_1(x) = x) & \forall x(f_m(f_{m'}(x)) = f_{mm'}(x)) \\ \forall x(f_{m+m'}(x) = f_m(x) + f_{m'}(x)) & \forall x\,(f_0(x) = 0) \end{array}$$

In the second line of these axioms, 1 is the unit of the field k, and mm' refers to multiplication in k. In the third line, $m + m'$ refers to addition in k, and the 0 in $f_0(x)$ is the 0 in k. Note, that if the field k is infinite, there are infinitely many axioms to satisfy!

Exercise 10 The language L_k and the axioms for vector spaces given above, are not very satisfactory in the sense that there are many important things about vectors that cannot be expressed by L_k-formulas; for example, that x and y are linearly independent vectors.

Devise yourself a different language and different axioms which do allow you to express that two vectors are linearly independent over k. Mimicking the example of graphs, have two 1-place relation symbols S and V (for "scalar" and "vector" respectively). How do you express addition of vectors and scalar multiplication?

1.4.4 Basic Plane Geometry

The language L_{geom} of basic plane geometry has two 1-place relation symbols P and L for "point" and "line", and a 2-place relation symbol I for "point x lies on line y". The axioms are:

$$\begin{array}{l} \forall x(P(x) \vee L(x)) \\ \forall x \neg(P(x) \wedge L(x)) \\ \forall xy(I(x,y) \rightarrow (P(x) \wedge L(y))) \\ \forall xx'(P(x) \wedge P(x') \rightarrow \exists y(I(x,y) \wedge I(x',y))) \\ \forall xx'yy'((I(x,y) \wedge I(x',y) \wedge I(x,y') \wedge I(x',y')) \rightarrow (x = x' \vee y = y')) \end{array}$$

Convince yourself that these axioms mean: everything is either a point or a line (and not both), for every two points there is a line they lie on, and two distinct lines can have at most one point in common.

Exercise 11 A famous extra axiom says, that for every line l and point x not on l, there is a unique line m through x, which does not intersect l. Show how to express this axiom in L_{geom}.

We conclude this chapter with a couple of definitions, for further reference.

Definition 1.8 Let L be a language. A *theory* in L, or an L-*theory*, is a set of L-sentences. If Γ is an L-theory, a *model* of Γ is an L-structure M such that $M \models \phi$ for every $\phi \in \Gamma$. An L-theory Γ is called *consistent* if Γ has a model. Furthermore we have the following notation: $\Gamma \models \phi$ means that $M \models \phi$ for every model M of Γ.

Exercise 12 Prove that an L-theory Γ is consistent if and only if $\Gamma \not\models \bot$.

Definition 1.9 An L-theory Γ is called *complete* if for every L-sentence ϕ we have $\Gamma \models \phi$ or $\Gamma \models \neg\phi$. If Γ is not complete and ϕ is an L-sentence such that $\Gamma \not\models \phi$ and $\Gamma \not\models \neg\phi$, we call ϕ *independent of* Γ.

Exercise 13 An L-theory Γ is consistent and complete precisely when there is an L-structure M such that

$$\Gamma \ = \ \{\phi \,|\, M \models \phi\}$$

Chapter 2

Proofs

In Chapter 2, we have introduced languages and formulas as mathematical objects: formulas are just certain finite sequences of elements of a certain set. Given a specific model, such formulas become mathematical statements via the definition of truth in that model.

In mathematical reasoning, one often observes that one statement "follows" from another, without reference to specific models or truth, as a purely "logical" inference. More generally, statements can be conjectures, assumptions or intermediate conclusions in a mathematical argument.

In this chapter we shall give a formal, abstract definition of a concept called 'proof'. A proof will be a finite object which has a number of *assumptions* which are formulas, and a *conclusion* which is a formula. Given a fixed language L, there will be a set of all proofs in L, and we shall be able to prove the *Completeness Theorem*:

> For a set Γ of L-sentences and an L-sentence ϕ, the relation $\Gamma \models \phi$ holds if and only if there exists a proof in L with conclusion ϕ and assumptions from the set Γ.

Recall that $\Gamma \models \phi$ was defined as: for every L-structure M which is a model of Γ, it holds that $M \models \phi$.

Therefore, the Completeness Theorem reduces a universal ("for all") statement about a large class of structures, to an existential ("there is") statement about one set (the set of proofs). Furthermore, we shall see that proofs are built up by rules that can be interpreted as elementary reasoning steps (we shall not go into the philosophical significance of this). Finally, we wish to remark that it can be effectively tested whether or not an object of appropriate kind is a 'proof', and that the set of all sentences ϕ such that

$\Gamma \models \phi$ can be effectively generated by a computer (we refer to the lecture course in Recursion Theory for a precise meaning of this).

2.1 Proof Trees

In a well-structured mathematical argument, it is clear at every point what the conclusion reached so far is, what the current assumptions are and on which intermediate results each step depends.

We model this mathematically with the concept of a *tree*.

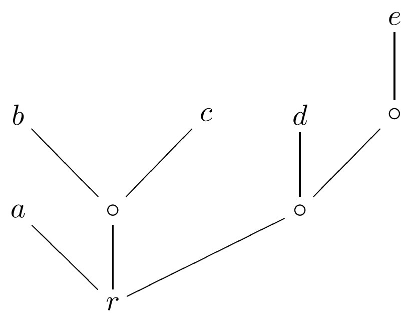
Definition 2.1 A *tree* is a partial order $(T, \leq)$ which has a least element, and is such that for every $x \in T$, the set

$$\downarrow(x) \ \equiv \ \{y \in T \,|\, y \leq x\}$$

is well-ordered by the relation $\leq$.

We shall only be concerned with *finite* trees; that is, finite posets T with least element, such that each $\downarrow(x)$ is linearly ordered.

This is an example of a tree:



We use the following dendrological language when dealing with trees: the least element is called the *root* (in the example above, the element marked r), and the maximal elements are called the *leaves* (in the example, the elements marked a, b, c, d, e).

When we see a proof as a tree, the leaves are the places for the assumptions, and the root is the place for the conclusion. The information that the assumptions give, may be compared to the carbon dioxide in real trees, which finds its way from the leaves to the root.

The following exercise gives some alternative ways of characterizing trees.

Exercise 14 a) Show that a finite tree is the same thing as a finite sequence of nonempty finite sets and functions

$$A_n \to \cdots \to A_1 \to A_0$$

where A_0 is a one-element set.

b) Show that a finite tree is the same thing as a finite set V together with a function $f : V \to V$ which has the properties that f has exactly one fixed point $r = f(r)$, and there are no elements $x \neq r$ such that $x = f^n(x)$ for some $n \in \mathbb{N}$.

c) If V is a finite set, a *hierarchy* on V is a collection $\mathcal{C}$ of subsets of V, such that $V \in \mathcal{C}$, and for any two elements $C_1 \neq C_2$ of $\mathcal{C}$, we have $C_1 \subset C_2$ or $C_2 \subset C_1$ or $C_1 \cap C_2 = \emptyset$. Let us call $\mathcal{C}$ a T_1-*hierarchy* if for each $x, y \in V$ with $x \neq y$, there is $C \in \mathcal{C}$ such that either $x \in C$ and $y \notin C$, or $y \in C$ and $x \notin C$. Call $\mathcal{C}$ *connected* if there is an element $r \in V$ such that the only element $C \in \mathcal{C}$ such that $r \in C$, is V itself.

Show that a finite tree is the same thing as a finite set V together with a connected T_1-hierarchy on V.

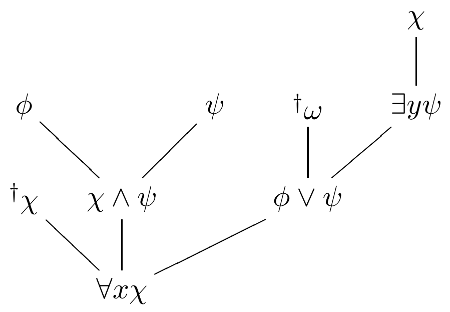
d) Let $\mathcal{B}$ be a set of finite trees such that for every finite tree there is exactly one element of $\mathcal{B}$ which is isomorphic to it. Let L be the language with one constant c and for every $n \in \mathbb{N}_{\geq 1}$ exactly one function symbol F_n of arity n. Show that $\mathcal{B}$ can be made into an L-structure such that for every other L-structure M and every $m \in M$, there is a unique function $f : \mathcal{B} \to M$ with the properties:

i) $f(c) = m$, and

ii) for all $n \in \mathbb{N}_{\geq 1}$ and every n-tuple $(b_1, \ldots, b_n)$ from $\mathcal{B}$,

$$f(F_n(b_1, \ldots, b_n)) = (F_n)^M(f(b_1), \ldots, f(b_n))$$

We shall be interested in L-*labelled* trees; that is: trees where the elements have 'names' which are L-formulas or formulas marked with a symbol †. For example:



The following definition formalizes this:

Definition 2.2 Let L be a language. We fix an extra symbol $\dagger$. A *marked* L-formula is a pair $(\dagger, \varphi)$; we shall write $^\dagger\varphi$ for $(\dagger, \varphi)$. Let $F(L)$ be the set of L-formulas, and let $^\dagger F(L)$ be the disjoint union of $F(L)$ and the set $\{\dagger\} \times F(L)$ of marked L-formulas.

An L-*labelled* tree is a finite tree T together with a function f from T to the set $^\dagger F(L)$, such that the only elements x of T such that $f(x)$ is a marked formula, are leaves of T.

The function f is called the *labelling function*, and $f(x)$ is called the *label* of x.

Among the L-labelled trees, we shall single out a set of 'proof trees'. The definition (Definition 2.3 below) uses the following two operations on L-labelled trees:

1). *Joining a number of labelled trees by adding a new root labelled* ϕ
Suppose we have a finite number of labelled trees $T_1, \ldots, T_k$ with labelling functions $f_1, \ldots, f_k$. Let T be the disjoint union $T_1 + \cdots + T_k$ together with a new element r, and ordered as follows: $x \leq y$ if and only if $x = r$, or for some i, $x, y \in T_i$ and $x \leq y$ holds in T_i.

Let the labelling function f on T be such that it extends each f_i on T_i and has $f(r) = \phi$.

We denote this construction by $\Sigma(T_1, \ldots, T_k; \phi)$.

2). *Adding some markings*
Suppose T is a labelled tree with labelling function f. If V is a set of leaves of T, we may modify f to f' as follows: $f'(x) = f(x)$ if $x \notin V$ or $f(x)$ is a marked formula; otherwise, $f'(x) = (\dagger, f(x))$.

We denote this construction by $Mk(T; V)$.

Exercise 15 Show that every L-labelled tree can be constructed by a finite number of applications of these two constructions, starting from one element trees with unmarked labels.

For the rest of this section, we shall assume that we have a fixed language L which we won't mention (we say 'labelled' and 'formula' instead of 'L-labelled', 'L-formula' etc.). Let us also repeat that for us from now on, 'tree' means *finite* tree.

If T is a labelled tree with labelling function f, root r and leaves $a_1, \dots, a_n$, we shall call the formula $f(r)$ (if it is a formula, that is: unmarked) the *conclusion* of T and the formulas $f(a_i)$ the *assumptions* of T. Assumptions of the form $^\dagger\varphi$ are called *eliminated assumptions.*

We can now give the promised definition of 'proof tree'. Instead of reading through the definition in one go, the reader is advised to work through a few clauses, and then have a look at the examples given after the definition; referring back to it when necessary.

Definition 2.3 The set $\mathcal{P}$ of *proof trees* is the smallest set of labelled trees, satisfying:

Ass For every formula φ, the tree with one element r and labelling function $f(r) = \varphi$, is an element of $\mathcal{P}$. Note that φ is both assumption and conclusion of this tree. We call this tree an *assumption tree.*

$\wedge I$ If T_1 and T_2 are elements of $\mathcal{P}$ with conclusions φ_1 and φ_2 respectively, then $\Sigma(T_1, T_2; \varphi_1 \wedge \varphi_2)$ is an element of $\mathcal{P}$. We say this tree was formed by $\wedge$-*introduction.*

$\wedge E$ If T is an element of $\mathcal{P}$ with conclusion $\phi \wedge \psi$ then both $\Sigma(T; \phi)$ and $\Sigma(T; \psi)$ are elements of $\mathcal{P}$. These are said to be formed by $\wedge$-*elimination.*

$\vee I$ If T is an element of $\mathcal{P}$ with conclusion φ, and ψ is any formula, then both $\Sigma(T; \varphi \vee \psi)$ and $\Sigma(T; \psi \vee \varphi)$ are elements of $\mathcal{P}$. We say these are formed by $\vee$-introduction.

$\vee E$ Suppose that T, S_1, S_2 are elements of $\mathcal{P}$ such that the conclusion of T is $\varphi \vee \psi$ and the conclusions of S_1 and S_2 are the same (say, χ). Let V_1 be the subset of the leaves of S_1 labelled φ, and let V_2 be the subset of the leaves of S_2 labelled ψ. Let $S_1' = Mk(S_1; V_1)$, $S_2' = Mk(S_2; V_2)$. Then $\Sigma(T, S_1', S_2'; \chi)$ is an element of $\mathcal{P}$ ($\vee$-elimination).

$\to I$ Suppose T is an element of $\mathcal{P}$ with conclusion φ, and let ψ be any formula. Let V be the subset of the set of leaves of T with label ψ, and $T' = Mk(T;V)$. Then $\Sigma(T';\psi \to \varphi)$ is an element of $\mathcal{P}$ ($\to$-introduction).

$\to E$ Suppose T and S are elements of $\mathcal{P}$ with conclusions $\varphi \to \psi$ and φ, respectively. Then $\Sigma(T,S;\psi)$ is an element of $\mathcal{P}$ ($\to$-elimination).

$\neg I$ Suppose T is an element of $\mathcal{P}$ with conclusion $\bot$. Let φ be any formula, and V be the subset of the set of leaves of T labelled φ. Let $T' = Mk(T;V)$. Then $\Sigma(T';\neg\varphi)$ is an element of $\mathcal{P}$ ($\neg$-introduction).

$\neg E$ Suppose T and S are elements of $\mathcal{P}$ with conclusions φ and $\neg\varphi$, respectively. Then $\Sigma(T,S;\bot)$ is an element of $\mathcal{P}$ ($\neg$-elimination).

$\bot E$ Suppose T is an element of $\mathcal{P}$ with conclusion $\bot$. Let φ be any formula, and V the subset of the set of leaves of T labelled $\neg\varphi$. Let $T' = Mk(T;V)$. Then $\Sigma(T';\varphi)$ is an element of $\mathcal{P}$ ($\bot$-elimination; one also hears *reductio ad absurdum* or *proof by contradiction*).

Subst Suppose T and S are elements of $\mathcal{P}$ such that the conclusion of T is $\varphi[t/x]$ and the conclusion of S is $(t = s)$. Suppose furthermore that the substitutions $\varphi[t/x]$ and $\varphi[s/x]$ are defined (recall from Chapter 2: this means that no variable in t or s becomes bound in the substitution). Then $\Sigma(T,S;\varphi[s/x])$ is an element of $\mathcal{P}$ (Substitution).

$\forall I$ Suppose T is an element of $\mathcal{P}$ with conclusion $\varphi[u/v]$, where u is a variable which does not occur in any unmarked assumption of T or in the formula $\forall v\varphi$ (and is not bound in φ). Then $\Sigma(T;\forall v\varphi)$ is an element of $\mathcal{P}$ ($\forall$-introduction).

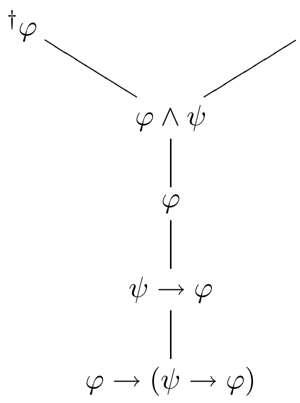
$\forall E$ Suppose T is an element of $\mathcal{P}$ with conclusion $\forall u\varphi$, and t is a term such that the substitution $\varphi[t/u]$ is defined. Then $\Sigma(T;\varphi[t/u])$ is an element of $\mathcal{P}$ ($\forall$-elimination).

$\exists I$ Suppose T is an element of $\mathcal{P}$ with conclusion $\varphi[t/u]$, and suppose the substitution $\varphi[t/u]$ is defined. Then $\Sigma(T;\exists u\varphi)$ is an element of $\mathcal{P}$ ($\exists$-introduction).

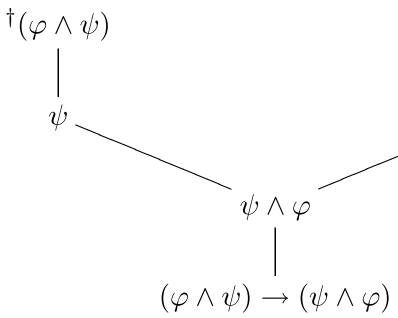
$\exists E$ Suppose T and S are elements of $\mathcal{P}$ with conclusions $\exists x\varphi$ and χ, respectively. Let u be a variable which doesn't occur in φ or χ, and is such that the only unmarked assumptions of S in which u occurs, are of the form $\varphi[u/x]$. Let V be the set of leaves of S with label $\varphi[u/x]$, and $S' = Mk(S;V)$. Then $\Sigma(T,S';\chi)$ is an element of $\mathcal{P}$ ($\exists$-elimination).

Examples. The following labelled trees are proof trees. Convince yourself of this, and find out at which stage labels have been marked:

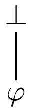
a)



b)

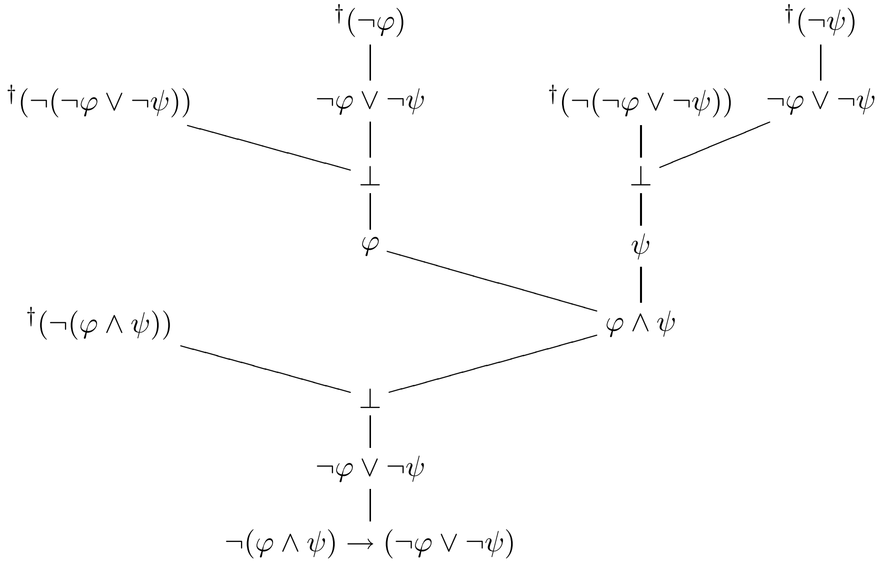

$^\dagger(\varphi \wedge \psi)$
$^\dagger(\varphi \wedge \psi)$
ψ
φ
$\psi \wedge \varphi$
$(\varphi \wedge \psi) \to (\psi \wedge \varphi)$

c)

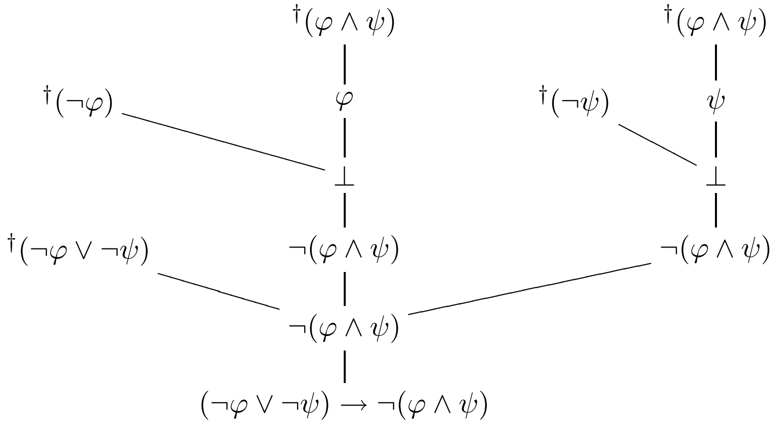


“Ex falso sequitur quodlibet”

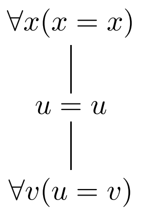
d)


$^\dagger(\neg\varphi)$
$^\dagger(\neg\psi)$
$^\dagger(\neg(\neg\varphi \vee \neg\psi))$
$\neg\varphi \vee \neg\psi$
$^\dagger(\neg(\neg\varphi \vee \neg\psi))$
$\neg\varphi \vee \neg\psi$
$\perp$
$\perp$
φ
ψ
$^\dagger(\neg(\varphi \wedge \psi))$
$\varphi \wedge \psi$
$\perp$
$\neg\varphi \vee \neg\psi$
$\neg(\varphi \wedge \psi) \rightarrow (\neg\varphi \vee \neg\psi)$

e)


$^\dagger(\varphi \wedge \psi)$
$^\dagger(\varphi \wedge \psi)$
$^\dagger(\neg\varphi)$
φ
$^\dagger(\neg\psi)$
ψ
$\perp$
$\perp$
$^\dagger(\neg\varphi \vee \neg\psi)$
$\neg(\varphi \wedge \psi)$
$\neg(\varphi \wedge \psi)$
$\neg(\varphi \wedge \psi)$
$(\neg\varphi \vee \neg\psi) \rightarrow \neg(\varphi \wedge \psi)$

f) The following "example" illustrates why, in formulating the rule $\forall I$, we have required that the variable u does not occur in the formula $\forall v\varphi$. For, let φ be the formula $u = v$. Consider that $(u = v)[u/v]$ is $u = u$, so were it not for this requirement, the following tree would be a valid proof tree:



Clearly, we would not like to accept this as a valid proof!

Definition 2.4 We define the relation

$$\Gamma \vdash \varphi$$

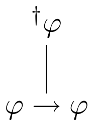
as: there is a proof tree with conclusion φ and whose unmarked assumptions are either elements of Γ or of the form $\forall x(x = x)$ for some variable x. We abbreviate $\{\varphi\} \vdash \psi$ as $\varphi \vdash \psi$, we write $\vdash \psi$ for $\emptyset \vdash \psi$, and $\Gamma, \varphi \vdash \psi$ for $\Gamma \cup \{\varphi\} \vdash \psi$.

Exercise 16 (Deduction Theorem) Prove, that the relation $\Gamma, \varphi \vdash \psi$ is equivalent to $\Gamma \vdash \varphi \rightarrow \psi$.

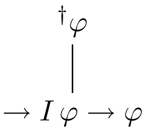
2.1.1 Variations and Examples

One variation in the notation of proof trees is, to write the name of each construction step next to the labels in the proof tree.

For example, the proof tree



is constructed from the assumption tree φ by →-introduction (at which moment the assumption φ is marked). One could make this explicit by writing



Another notational variation is one that is common in the literature: the ordering is indicated by horizontal bars instead of vertical or skew lines, and next to these bars, it is indicated by which of the constructions of Definition 2.3, the new tree results from the old one(s). Assumptions are numbered, such that different assumptions have different numbers, but distinct occurrences of the same assumption may get the same number. If, in the construction, assumptions are marked, this is indicated by their numbers next to the name of the construction.

In this style, the proof tree

$$\begin{array}{c} {}^{\dagger}\varphi \\ | \\ \varphi \to \varphi \end{array}$$

looks as follows:

$$\dfrac{{}^{\dagger}\varphi^{1}}{\varphi \to \varphi} \to I, 1$$

We shall call this a *decorated proof tree*. Although (or maybe: because!) they contain some redundant material, decorated proof trees are easier to read and better suited to practise the construction of proof trees.

In decorated style, examples a)–e) of the previous section are as follows:

a)

$$\dfrac{\dfrac{\dfrac{\dfrac{{}^{\dagger}\varphi^{1} \qquad {}^{\dagger}\psi^{2}}{\varphi \wedge \psi} \wedge I}{\varphi} \wedge E}{\psi \to \varphi} \to I, 2}{\varphi \to (\psi \to \varphi)} \to I, 1$$

The assumption φ, numbered 1, gets marked when construction $\to I$ with number 1 is performed; etc.

b)

$$\dfrac{\dfrac{\dfrac{{}^{\dagger}\varphi \wedge \psi^{1}}{\psi} \wedge E \qquad \dfrac{{}^{\dagger}\varphi \wedge \psi^{1}}{\varphi} \wedge E}{\psi \wedge \varphi} \wedge I}{(\varphi \wedge \psi) \to (\psi \wedge \varphi)} \to I, 1$$

c)

$$\dfrac{\bot}{\varphi} \bot E$$

d)

$$\dfrac{\dfrac{\dfrac{{}^\dagger\neg(\varphi\wedge\psi)^4 \qquad \dfrac{\dfrac{\dfrac{{}^\dagger\neg(\neg\varphi\vee\neg\psi)^3 \quad \dfrac{{}^\dagger\neg\varphi^1}{\neg\varphi\vee\neg\psi}\vee I}{\bot}\neg E}{\varphi}\bot E,1 \qquad \dfrac{\dfrac{{}^\dagger\neg(\neg\varphi\vee\neg\psi)^3 \quad \dfrac{{}^\dagger\neg\psi^2}{\neg\varphi\vee\neg\psi}\vee I}{\bot}\neg E}{\psi}\bot E,2}{\varphi\wedge\psi}\wedge I}{\bot}\neg E}{\neg\varphi\vee\neg\psi}\bot E,3}{\neg(\varphi\wedge\psi)\to(\neg\varphi\vee\neg\psi)}\to I,4$$

e)

$$\dfrac{\dfrac{{}^\dagger\neg\varphi\vee\neg\psi^5 \qquad \dfrac{\dfrac{{}^\dagger\neg\varphi^3 \quad \dfrac{{}^\dagger\varphi\wedge\psi^1}{\varphi}\wedge E}{\bot}\neg E}{\neg(\varphi\wedge\psi)}\neg I,1 \qquad \dfrac{\dfrac{{}^\dagger\neg\psi^4 \quad \dfrac{{}^\dagger\varphi\wedge\psi^2}{\psi}\wedge E}{\bot}\neg E}{\neg(\varphi\wedge\psi)}\neg I,2}{\neg(\varphi\wedge\psi)}\vee E,3,4}{(\neg\varphi\vee\neg\psi)\to\neg(\varphi\wedge\psi)}\to I,5$$

Some more examples:

f) A proof tree for $t = s \vdash s = t$:

$$\dfrac{\dfrac{\forall x(x=x)}{t=t}\forall E \qquad t=s}{s=t}\text{Subst}$$

The use of Subtitution is justified since $t = t$ is $(u = t)[t/u]$. Quite similarly, we have a proof tree for $\{t = s, s = r\} \vdash t = r$:

$$\dfrac{t=s \qquad s=r}{t=r}\text{Subst}$$

g)

$$\dfrac{\dfrac{\dfrac{\dfrac{{}^\dagger\neg\exists x\varphi(x)^2 \qquad \dfrac{{}^\dagger\varphi(y)^1}{\exists x\varphi(x)}\exists I}{\bot}\neg E}{\neg\varphi(y)}\neg I,1}{\forall x\neg\varphi(x)}\forall I}{\neg\exists x\varphi(x)\to\forall x\neg\varphi(x)}\to I,2$$

You should check why application of $\forall I$ is justified in this tree.

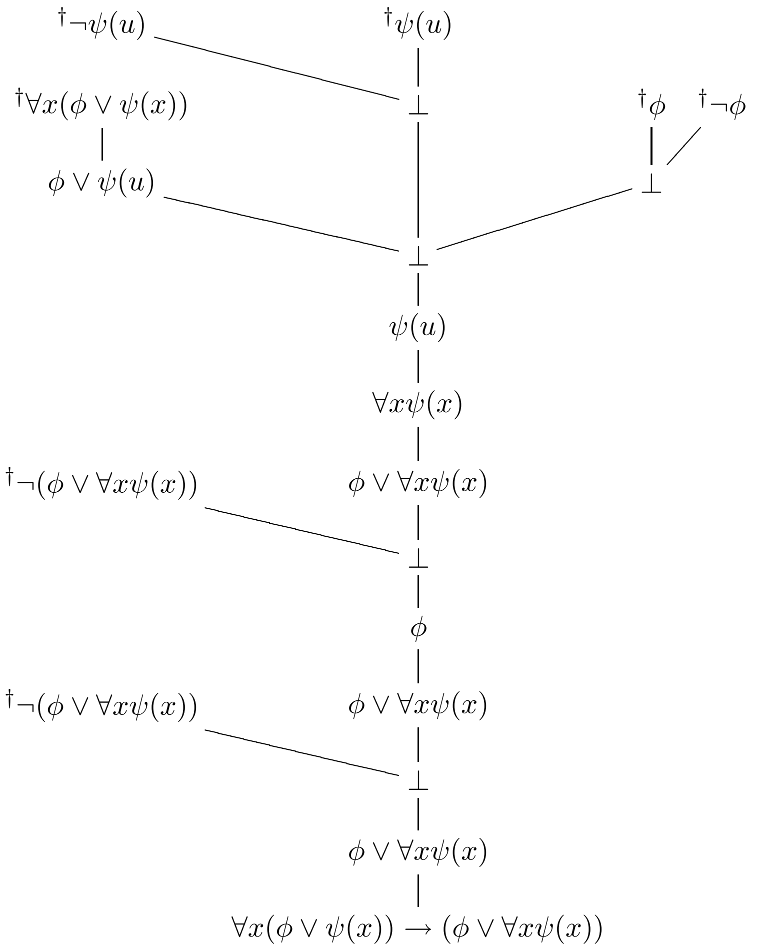
h) The following tree gives an example of the $\exists E$-construction:

$$\dfrac{\dfrac{\dfrac{{}^{\dagger}\exists x\varphi(x)^2 \qquad \dfrac{\dfrac{{}^{\dagger}\forall x\neg\varphi(x)^3}{\neg\varphi(y)}\,\forall E \qquad {}^{\dagger}\varphi(y)^1}{\bot}\,\neg E}{\bot}\,\exists E,1}{\neg\exists x\varphi(x)}\,\neg I,2}{\forall x\neg\varphi(x)\to\neg\exists x\varphi(x)}\,\to I,3$$

i)

$$\dfrac{\dfrac{\dfrac{{}^{\dagger}\neg\forall x\neg\varphi(x)^3 \qquad \dfrac{\dfrac{\dfrac{{}^{\dagger}\neg\exists x\varphi(x)^2 \qquad \dfrac{{}^{\dagger}\varphi(y)^1}{\exists x\varphi(x)}\,\exists I}{\bot}\,\neg E}{\neg\varphi(y)}\,\neg I,1}{\forall x\neg\varphi(x)}\,\forall I}{\bot}\,\neg E}{\exists x\varphi(x)}\,\bot E,2}{\neg\forall x\neg\varphi(x)\to\exists x\varphi(x)}\,\to I,2$$

j) The following tree is given in undecorated style; it is a good exercise to decorate it. It is assumed that the variable x does not occur in ϕ; check that without this condition, it is not a correct proof tree:


$^\dagger\neg\psi(u)$
$^\dagger\psi(u)$
$^\dagger\forall x(\phi \vee \psi(x))$
$\bot$
$^\dagger\phi$
$^\dagger\neg\phi$
$\phi \vee \psi(u)$
$\bot$
$\bot$
$\psi(u)$
$\forall x\psi(x)$
$^\dagger\neg(\phi \vee \forall x\psi(x))$
$\phi \vee \forall x\psi(x)$
$\bot$
ϕ
$^\dagger\neg(\phi \vee \forall x\psi(x))$
$\phi \vee \forall x\psi(x)$
$\bot$
$\phi \vee \forall x\psi(x)$
$\forall x(\phi \vee \psi(x)) \to (\phi \vee \forall x\psi(x))$

A bit of heuristics. When faced with the problem of constructing a proof tree which has a specified set of unmarked assumptions Γ and a prescribed conclusion ϕ (often formulated as: "construct a proof tree for $\Gamma \vdash \phi$"), it is advisable to use the following heuristics (but there is no guarantee that they work! Or, that they produce the most efficient proof):

> If ϕ is a conjunction $\phi_1 \wedge \phi_2$, break up the problem into two problems $\Gamma \vdash \phi_1$ and $\Gamma \vdash \phi_2$;
>
> If ϕ is an implication $\phi_1 \to \phi_2$, transform the problem into $\Gamma \cup \{\phi_1\} \vdash \phi_2$;
>
> If ϕ is a negation $\neg\psi$, transform into $\Gamma \cup \{\psi\} \vdash \bot$;
>
> If ϕ is of form $\forall x\psi(x)$, transform into $\Gamma \vdash \psi(u)$;

In all other (non-obvious) cases, try $\Gamma \cup \{\neg\phi\} \vdash \bot$.

Exercise 17 Construct proof trees for the equivalences of Exercise 2. Recall that $\leftrightarrow$ is an abbreviation: for example, a proof tree for $\vdash (\varphi \to \psi) \leftrightarrow (\neg\varphi \vee \psi)$ will be constructed out of two proof trees, one for $\{\varphi \to \psi \vdash \neg\varphi \vee \psi$, and one for $\neg\varphi \vee \psi \vdash \varphi \to \psi$, by applying $\to$- and $\wedge$-introduction.

2.1.2 Induction on Proof Trees

Since the set $\mathcal{P}$ of proof trees is defined as the *least* set of labelled trees which contains the assumption tree φ and is closed under a number of constructions (definition 2.3), $\mathcal{P}$ is susceptible to proofs by *induction* over proof trees: if $\mathcal{A}$ is any set of labelled trees which contains φ and is closed under the constructions, then $\mathcal{A}$ contains $\mathcal{P}$ as a subset.

Some examples of properties of proof trees one can prove by this method:

1. No proof tree has a marked formula at the root.
2. In every proof tree T, for every $x \in T$ there are at most 3 elements of T directly above x (we say that every proof tree is a *ternary tree*).
3. If T is a proof tree such that the conclusion of T is of the form $\varphi[c/u]$, where c is a constant that does not occur in any unmarked assumption of T, and v is a variable which doesn't occur anywhere in T, then replacing c by v throughout in T, results in a new proof tree.

In the proof of the Soundness Theorem (section 2.2 below) we shall also apply induction over proof trees.

Exercise 18 Let $\Gamma \vdash_H \varphi$ be defined as the least relation between sets of L-formulas Γ and L-formulas φ, such that the following conditions are satisfied:

i) If $\varphi \in \Gamma$, then $\Gamma \vdash_H \varphi$;

ii) if $\Gamma \vdash_H \varphi$ and $\Gamma \vdash_H \psi$ then $\Gamma \vdash_H (\varphi \wedge \psi)$, and conversely;

iii) if $\Gamma \vdash_H \varphi$ or $\Gamma \vdash_H \psi$, then $\Gamma \vdash_H (\varphi \vee \psi)$;

iv) if $\Gamma \cup \{\varphi\} \vdash_H \chi$ and $\Gamma \cup \{\psi\} \vdash_H \chi$, then $\Gamma \cup \{\varphi \vee \psi\} \vdash_H \chi$;

v) if $\Gamma \cup \{\varphi\} \vdash \bot$, then $\Gamma \vdash_H \neg\varphi$;

vi) if $\Gamma \vdash_H \varphi$ and $\Gamma \vdash_H \neg\varphi$ then $\Gamma \vdash_H \bot$;

vii) if $\Gamma \cup \{\neg\varphi\} \vdash_H \bot$ then $\Gamma \vdash_H \varphi$;

viii) if $\Gamma \cup \{\varphi\} \vdash_H \psi$ then $\Gamma \vdash_H \varphi \to \psi$;

ix) if $\Gamma \vdash_H \varphi$ and $\Gamma \vdash_H \varphi \to \psi$ then $\Gamma \vdash_H \psi$;

x) if $\Gamma \vdash_H \psi(u)$ and u does not occur in Γ, then $\Gamma \vdash_H \forall x \psi(x)$;

xi) if $\Gamma \vdash_H \forall x \psi(x)$ then if $\psi[t/x]$ is defined, $\Gamma \vdash_H \psi[t/x]$;

xii) if $\psi[t/x]$ is defined and $\Gamma \vdash_H \psi[t/x]$, then $\Gamma \vdash_H \exists x \psi(x)$;

xiii) if $\Gamma \cup \{\psi(u)\} \vdash_H \chi$ and u does not occur in Γ or χ, then $\Gamma \cup \{\exists x \psi(x)\} \vdash_H \chi$.

Show that the relation $\Gamma \vdash_H \varphi$ coincides with the relation $\Gamma \vdash \varphi$ from Definition 2.4.

2.2 Soundness and Completeness

We compare the relation $\Gamma \vdash \phi$ from Definition 2.4 to the relation $\Gamma \models \phi$ from definition 1.8 in Chapter 1; recall that the latter means: in every model M of Γ, the sentence ϕ holds.

Here we just state the two fundamental theorems of Logic, for a set of sentences Γ and a sentence ϕ :

Theorem 2.5 (Soundness Theorem) *If* $\Gamma \vdash \phi$ *then* $\Gamma \models \phi$.

Theorem 2.6 (Completeness Theorem; Gödel, 1930) *If* $\Gamma \models \phi$ *then* $\Gamma \vdash \phi$.

Exercise 19 Prove that Theorems 2.5–2.6 together are equivalent to the statement: let Γ be a theory. The Γ is consistent if and only if $\Gamma \not\vdash \bot$.

Exercise 20 [Compactness Theorem; Gödel 1930] Prove from Theorems 2.5–2.6 the Compactness Theorem: if every finite subset of a theory Γ is consistent, then Γ is consistent.

Conclude from this the following equivalent formulation: if Γ is a theory in a language L, and ϕ is an L-sentence such that $\Gamma \models \phi$, then there is a finite subset Γ' of Γ such that $\Gamma' \models \phi$.

2.3 Extensions of Theories by Defined Notions

If one is to write out a real mathematical proof of an interesting theorem as a proof tree, then almost always the formulas become way too long to be readable. Therefore, we are led to make abbreviations, but also, to introduce new function symbols and relation symbols for 'defined' functions and relations. However, when we want to say something about a fixed theory Γ, we need to make sure that if we enlarge the language with new function and relation symbols, and enlarge the theory with axioms about these new symbols, we still have a theory which is 'close enough' to Γ, in the sense of the following definition.

Definition 2.7 Let L and L' be two languages such that $L \subset L'$; let Γ be an L-theory and Γ' be an L'-theory such that $\Gamma \subset \Gamma'$. Then Γ' is said to be a *conservative extension* of Γ, if for every L-sentence ϕ such that $\Gamma' \models \phi$, it already holds that $\Gamma \models \phi$.

Exercise 21 Suppose we have a chain of languages $L_0 \subset L_1 \subset \cdots$, and for every i we have an L_i-theory Γ_i, such that $\Gamma_0 \subset \Gamma_1 \subset \cdots$. Let $L = \bigcup_{i=0}^{\infty} L_i$ and $\Gamma = \bigcup_{i=0}^{\infty} \Gamma_i$.

Prove: if for every $i \geq 0$, Γ_{i+1} is a conservative extension of Γ_i, then Γ is a conservative extension of Γ_0.

[Hint: use the Compactness Theorem]

A very common way to construct conservative extensions is the introduction of *Skolem functions*. Suppose Γ is an L-theory and $\varphi(x_1, \ldots, x_k, y)$ is an L-formula with free variables $x_1, \ldots, x_k, y$. Suppose:

$$\Gamma \models \forall x_1 \cdots \forall x_k \exists y \varphi(x_1, \ldots, x_k, y)$$

Now let F be a new k-place function symbol, not in L. Let L' be $L \cup \{F\}$, and Γ' be the L'-theory defined by

$$\Gamma' \ = \ \Gamma \cup \{\forall x_1 \cdots \forall x_k \varphi(x_1, \ldots, x_k, F(x_1, \ldots, x_k))\}$$

Then Γ' is a conservative extension of Γ (we also say Γ' is *conservative over* Γ). This can be seen as follows: suppose ψ is an L-sentence such that $\Gamma' \models \psi$. We need to prove that $\Gamma \models \psi$. To this end, let M be a model of Γ. Then we have:

$$M \models \forall x_1 \cdots \forall x_k \exists y \varphi(x_1, \ldots, x_k, y)$$

so for every k-tuple $m_1, \ldots, m_k$ of elements of M we can find an element n of M such that $M \models \varphi(m_1, \ldots, m_k, n)$. That means, we can find a function

$f : M^n \to M$ such that for every k-tuple $m_1, \dots, m_k$ of elements of M, $M \models \varphi(m_1, \dots, m_k, f(m_1, \dots, m_k))$.

Now let M' be the L'-structure with the same underlying set as M, and the same interpretations of all the symbols from L, as M; and moreover, $F^{M'} = f$. Then clearly the following two statements are true:

i) M and M' satisfy the same L-sentences;

ii) M' is a model of Γ'.

From these two statements it follows immediately that $M \models \psi$, as desired.

Something similar can be done for relation symbols: for any formula φ with k free variables, one can (relative to an L-theory Γ, extend the language by one k-place relation symbol R_φ, and extend the theory by one axiom

$$\forall x_1 \cdots \forall x_k (R_\varphi(x_1, \dots, x_k) \leftrightarrow \varphi(x_1, \dots, x_k))$$

Then the new theory is conservative over Γ.

A special kind of conservative extensions are so-called *definitional extensions.*

Definition 2.8 Let $L \subset L'$, $\Gamma \subset \Gamma'$ be as in definition 2.7. Γ' is called a *definitional extension* of Γ if there is a function $(\cdot)^*$ from L'-sentences to L-sentences such that for every L'-sentence ϕ the following holds:

i) $\Gamma' \models \phi \leftrightarrow (\phi)^*$

ii) if $\Gamma' \models \phi$, then $\Gamma \models (\phi)^*$

iii) if ϕ is an L-sentence, then $\Gamma \models \phi \leftrightarrow (\phi)^*$

Exercise 22 Prove that every definitional extension is a conservative extension.

Exercise 23 Prove, in the situation of definition 2.8, that the function $(\cdot)^*$ preserves equivalence: if $\Gamma' \models \phi \leftrightarrow \psi$ then $\Gamma \models (\phi)^* \leftrightarrow (\psi)^*$.

Exercise 24 Let L_i, Γ_i, L and Γ be as in exercise 21. Prove: if for every i, Γ_i is a definitional extension of Γ_i, then Γ is a definitional extension of Γ_0.

An example of a definitional extension arises if we introduce Skolem functions for uniquely defined elements. Let us introduce an important notation. **Notation**. The quantifier $\exists ! x \cdots$ means: there is *exactly one* x such that.... So the expression $\exists ! x \varphi(x)$ can be seen as an abbreviation for the formula

$$\exists x \forall u(\varphi(u) \leftrightarrow u = x)$$

Now suppose $\varphi(x_1, \ldots, x_k, y)$ is an L-formula such that

$$\Gamma \models \forall x_1 \cdots \forall x_k \exists ! y \varphi(x_1, \ldots, x_k, y)$$

Let Γ' be the extension of Γ by one Skolem function F for φ.

In this case, Γ' is a definitional extension of Γ. For an arbitrary L'-sentence ψ, we define $(\psi)^*$ inductively, by induction on the number of occurrences of F in ψ. If F does not occur in ψ, then ψ is an L-sentence, and we put $(\psi)^* = \psi$. Otherwise, there must be an occurrence of F in ψ such that F occurs in a term $F(t_1, \ldots, t_k)$ with all t_i's L-terms (not containing F). Take the first such occurrence, and let $\psi(u)$ be the formula resulting from ψ by replacing this occurrence of $F(t_1, \ldots, t_k)$ by the new variable u. Then define

$$(\psi)^* \;\equiv\; (\exists u(\varphi(t_1, \ldots, t_k, u) \wedge \psi(u)))^*$$

This is well-defined because the formula $\exists u(\varphi(t_1, \ldots, t_k, u) \wedge \psi(u))$ has one occurrence of F less than ψ.

Exercise 25 Show that $(\cdot)^*$ makes Γ' a definitional extension of Γ. Show also, that we might as well have put

$$(\psi)^* \;\equiv\; (\forall u(\varphi(t_1, \ldots, t_k, u) \rightarrow \psi(u)))^*$$

Chapter 3

(Primitive) Recursive Functions

3.1 Primitive recursive functions and relations

Notation for functions. In mathematical texts, it is common to use expressions containing variables, such as $x+y$, x^2, $x\log(y)$ etc., both for a (variable) *number* and for the *function* of the occurring variables: we say "the function $x+y$". However, when we are doing Logic and we think about ways of defining functions, it is better to distinguish these different meanings by different notations. The expression $x \log y$ may mean, for example:

- a real number
- a function of (x, y), that is a function: $\mathbb{R}^2 \to \mathbb{R}$
- a function of (y, x), i.e. another function: $\mathbb{R}^2 \to \mathbb{R}$
- a function of y (with parameter x, so actually a parametrized family of functions: $\mathbb{R} \to \mathbb{R}$)
- a function of (x, y, z), that is a function: $\mathbb{R}^3 \to \mathbb{R}$

In order to distinguish these meanings we employ the so-called λ-notation: if $\vec{x}$ is a sequence of variables $x_1 \cdots x_k$ which might occur in the expression G, then $\lambda\vec{x}.G$ denotes the function which assigns to the k-tuple $n_1 \cdots n_k$ the value $G(n_1, \ldots, n_k)$ (substitute the n_i for x_i in G). In this notation the 5 meanings above can be distuinguished by notation as follows: $x\log(y)$, $\lambda xy.x\log(y)$, $\lambda yx.x\log(y)$, $\lambda y.x\log(y)$ and $\lambda xyz.x\log(y)$.

Definition 3.1 The class of *primitive recursive* functions $\mathbb{N}^k \to \mathbb{N}$ (where k is allowed to vary over $\mathbb{N}$) is generated by the following clauses:

i) the *zero function* $Z = \lambda x.0$ is primitive recursive;

ii) the *successor function* $S = \lambda x.x + 1$ is primitive recursive;

iii) the *projections* $\Pi^k_i = \lambda x_1 \cdots x_k.x_i$ (for $1 \leq i \leq k$) are primitive recursive;

iv) If $G_1, \ldots, G_l : \mathbb{N}^k \to \mathbb{N}$ and $H : \mathbb{N}^l \to \mathbb{N}$ are primitive recursive, then so is

$$\lambda \vec{x}.H(G_1(\vec{x}), \ldots, G_l(\vec{x}))$$

this function is said to be defined from $G_1, \ldots, G_l$ and H by *composition*;

v) If $G : \mathbb{N}^k \to \mathbb{N}$ and $H : \mathbb{N}^{k+2} \to \mathbb{N}$ are primitive recursive, then so is the function F defined from G and H by *primitive recursion*:

$$\begin{array}{rcl} F(0, \vec{x}) & = & G(\vec{x}) \\ F(y + 1, \vec{x}) & = & H(y, F(y, \vec{x}), \vec{x}) \end{array}$$

Remark: in clause *v)* van definition 3.1 we *don't* exclude the case $k = 0$; in that case we take the definition to mean that if $H : \mathbb{N}^2 \to \mathbb{N}$ is primitive recursive and $n \in \mathbb{N}$, then the function F, defined by

$$\begin{array}{rcl} F(0) & = & n \\ F(y + 1) & = & H(y, F(y)) \end{array}$$

is also primitive recursive.

When we speak of a *k-ary relation*, we mean a subset of $\mathbb{N}^k$. We shall stick to the following convention for the *characteristic function* $\chi_A : \mathbb{N}^k \to \mathbb{N}$ of the k-ary relation A:

$$\chi_A(\vec{x})\ \chi_A(\vec{x}) = \begin{cases} 0 & \text{if } \vec{x} \in A \\ 1 & \text{else} \end{cases}$$

A relation is said to be primitive recursive if its characteristic function is.

Examples of primitive recursive functions. The following derivations show for a couple of simple functions that they are primitive recursive:

a) $\lambda xy.x+y$. For, $0+y=y=\Pi_1^1(y)$, and $(x+1)+y=S(x+y)=S(\Pi_2^3(x,x+y,y))$, hence $\lambda xy.x+y$ is defined by primitive recursion from Π_1^1 and a function defined by composition form S and Π_2^3;

b) $\lambda xy.xy$. For, $0y=0=Z(y)$, and $(x+1)y = xy+y = (\lambda xy.x+y)(\Pi_2^3(x,xy,y),\Pi_3^3(x,xy,y))$, hence $\lambda xy.xy$ is defined by primitive recursion from Z and a function defined by composition from $\lambda xy.x+y$ and projections;

c) $\lambda x.\mathrm{pd}(x)$ (the *predecessor function*: $\mathrm{pd}(x) = x-1$ if $x>0$, and $\mathrm{pd}(0)=0$). For, $\mathrm{pd}(0)=0$, and $\mathrm{pd}(x+1)=x=\Pi_1^2(x,\mathrm{pd}(x))$

Exercise 26 Prove that the following functions are primitive recursive:

i) $\lambda xy.x^y$

ii) $\lambda xy.x\dot{-}y$. This is *cut-off subtraction*: $x\dot{-}y = x-y$ if $x \geq y$, and $x\dot{-}y=0$ if $x<y$.

iii) $\lambda xy.\min(x,y)$

iv) sg (the *sign function*), where

$$\mathrm{sg}(x) = \begin{cases} 1 & \text{if } x>0 \\ 0 & \text{else} \end{cases}$$

v) $\overline{\mathrm{sg}}$, where

$$\overline{\mathrm{sg}}(x) = \begin{cases} 0 & \text{if } x>0 \\ 1 & \text{else} \end{cases}$$

vi) $\lambda xy.|x-y|$

vii) $\lambda x.n$ for fixed n

viii) $\lambda x.x!$

ix) $\lambda xy.\mathrm{rm}(x,y)$ where $\mathrm{rm}(x,y)=0$ if $y=0$, and the remainder of x on division by y otherwise.

Exercise 27 Prove that the following relations are primitive recursive:

i) $\{(x,y) \mid x = y\}$

ii) $\{(x,y) \mid x \leq y\}$

iii) $\{(x,y) \mid x|y\}$

iv) $\{x \mid x \text{ is a prime number}\}$

Exercise 28 Show that the function C is primitive recursive, where C is given by

$$C(x,y,z) = \begin{cases} x & \text{if } z = 0 \\ y & \text{else} \end{cases}$$

Therefore, we can define primitive recursive functions by 'cases', using primitive recursive relations.

Proposition 3.2

a) If the function $F : \mathbb{N}^{k+1} \to \mathbb{N}$ is primitive recursive, then so are the functions:

$$\begin{array}{l} \lambda \vec{x}z.\sum_{y<z} F(\vec{x},y) \\ \lambda \vec{x}z.\prod_{y<z} F(\vec{x},y) \\ \lambda \vec{x}z.(\mu y < z.F(\vec{x},y) = 0) \end{array}$$

The last of these is said to be defined from F by bounded minimization, *and produces the least $y < z$ for which $F(\vec{x},y) = 0$; if such an $y < z$ does not exist, it outputs z);*

b) If A and B are primitive recursive k-ary relations, then so are $A \cap B$, $A \cup B$, $A - B$ en $\mathbb{N}^k - A$;

c) If A is a primitive recursive $k+1$-ary relation, then the relations $\{(\vec{x},z) \mid \exists y < z(\vec{x},y) \in A\}$ and $\{(\vec{x},z) \mid \forall y < z(\vec{x},y) \in A\}$ are also primitive recursive.

Proof.

a) $\sum_{y<0} F(\vec{x},y) = 0$ and $\sum_{y<z+1} F(\vec{x},y) = \sum_{y<z} F(\vec{x},y) + F(\vec{x},z)$;
$\prod_{y<0} F(\vec{x},y) = 1$ and $\prod_{y<z+1} F(\vec{x},y) = (\prod_{y<z} F(\vec{x},y))F(\vec{x},z)$;
$(\mu y < 0.F(\vec{x},y) = 0) = 0$ and $(\mu y < z+1.F(\vec{x},y) = 0) = (\mu y < z.F(\vec{x},y) = 0) + \text{sg}(\prod_{y<z+1} F(\vec{x},y))$

b) $\chi_{A\cap B} = \lambda x.\text{sg}(\chi_A(x) + \chi_B(x))$
$\chi_{A\cup B} = \lambda x.\chi_A(x)\chi_B(x)$

Exercise 29 Finish the proof of Proposition 3.2.

∎

Exercise 30 If $F : \mathbb{N}^2 \to \mathbb{N}$ is primitive recursive, then so is $\lambda n. \sum_{k<n} F(n,k)$.

Proposition 3.3 *If G_1, G_2 and H are primitive recursive functions $\mathbb{N}^n \to \mathbb{N}$, then so is the function F, defined by*

$$F(\vec{x}) = \begin{cases} G_1(\vec{x}) & \textit{if } H(\vec{x}) = 0 \\ G_2(\vec{x}) & \textit{else} \end{cases}$$

Proof. For, $F(\vec{x}) = C(G_1(\vec{x}), G_2(\vec{x}), H(\vec{x}))$, where C is the function from exercise 28. ∎

Exercise 31 Let $p_0, p_1, \ldots$ be the sequence of prime numbers: $2, 3, 5, \ldots$ Show that the function $\lambda n.p_n$ is primitive recursive.

3.2 Coding of pairs and tuples

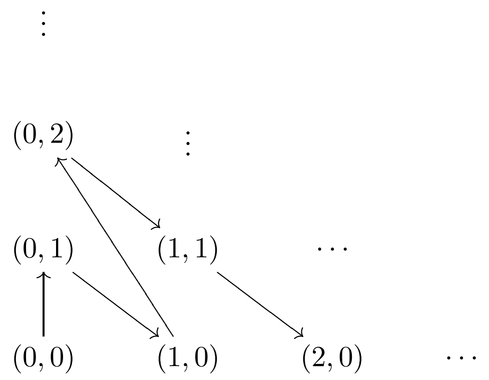
One of the basic ideas in Gödel's proof is that all kinds of structures (in particular: *terms*, *formulas* and *proofs*) can be *coded* as natural numbers. If a bit of care is taken with the coding, one can then also show that basic operations on these structures are given as primitive recursive functions on their codes. For example, if the code of a formula φ is denoted by $\ulcorner\varphi\urcorner$ and the code of the term t is $\ulcorner t\urcorner$ then there is a primitive recursive function F such that $F(\ulcorner\varphi\urcorner, \ulcorner t\urcorner) = \ulcorner\varphi[t/v]\urcorner$.

We shall have to code sequences of numbers as one number, in such a way that important operations on sequences, such as: taking the length of a sequence, the i'th element of the sequence, forming a sequence out of two sequences by putting one after the other (*concatenating* two sequences), are primitive recursive *in their codes.* This is carried out below.

Any bijection $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$ is called a *pairing function*: if $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ is bijective we say that $f(x,y)$ *codes the pair* (x,y). An example of such an f is the primitive recursive function $\lambda xy.2^x(2y+1) - 1$.

Exercise 32 Let $f(x,y) = 2^x(2y+1) - 1$. Prove that the functions $k_1 : \mathbb{N} \to \mathbb{N}$ and $k_2 : \mathbb{N} \to \mathbb{N}$ which satisfy $f(k_1(x), k_2(x)) = x$ for all x, are primitive recursive.

A simpler pairing function is given by the "diagonal enumeration" j of $\mathbb{N}\times\mathbb{N}$:



So, $j(0,0) = 0,\ j(0,1) = 1,\ j(1,0) = 2,\ j(0,2) = 3$ etc. We have:

$$j(n,m) = \#\{(k,l) \in \mathbb{N}\times\mathbb{N} \mid k+l < n+m\ \vee (k+l = n+m\ \wedge\ k < n)\}$$

in other words

$$j(n,m) = \frac{1}{2}(n+m)(n+m+1) + n = \frac{(n+m)^2 + 3n + m}{2}$$

The function j is given by a polynomial of degree 2. By the way, there is a theorem (the Fueter-Pólya Theorem, see [22]) which says that j and its 'twist' i.e. the function $\lambda nm.j(m,n)$ are the *only* polynomials of degree 2 that induce a bijection: $\mathbb{N}\times\mathbb{N} \to \mathbb{N}$.

It is convenient that $x \leq j(x,y)$ and $y \leq j(x,y)$, so if we define:

$$\begin{array}{l} j_1(z) = \mu x \leq z.[\exists y \leq z.j(x,y) = z] \\ j_2(z) = \mu y \leq z.[\exists x \leq z.j(x,y) = z] \end{array}$$

then $j(j_1(z), j_2(z)) = z$.

Exercise 33 Prove this and prove also that j_1 and j_2 are primitive recursive.

Exercise 34 (Simultaneous recursion) Suppose the functions $G_1,\ G_2 :\ \mathbb{N}^k \to \mathbb{N}$ and $H_1,\ H_2 :\ \mathbb{N}^{k+3} \to \mathbb{N}$ are primitive recursive. Define the functions F_1 and $F_2 :\ \mathbb{N}^{k+1} \to \mathbb{N}$ 'simultaneously' by the following scheme:

$$\begin{array}{rclrcl} F_1(0,\vec{x}) & = & G_1(\vec{x}) & F_1(y+1,\vec{x}) & = & H_1(y, F_1(y,\vec{x}), F_2(y,\vec{x}), \vec{x}) \\ F_2(0,\vec{x}) & = & G_2(\vec{x}) & F_2(y+1,\vec{x}) & = & H_2(y, F_1(y,\vec{x}), F_2(y,\vec{x}), \vec{x}) \end{array}$$

Check that F_1 en F_2 are well-defined, and use the pairing function j and its projections j_1 and j_2 to show that F_1 and F_2 are primitive recursive.

We are also interested in good bijections: $\mathbb{N}^n \to \mathbb{N}$ for $n > 2$. In general, such bijections can be given by polynomials of degree n, but we shall use polynomials of higher degree:

Definition 3.4 The bijections $j^m : \mathbb{N}^m \to \mathbb{N}$ for $m \geq 1$ are defined by:

$$\begin{array}{rl} j^1 & \text{is the identity function} \\ j^{m+1}(x_1, \ldots, x_m, x_{m+1}) & = j(j^m(x_1, \ldots, x_m), x_{m+1}) \end{array}$$

Then we also have *projection functions* $j_i^m : \mathbb{N} \to \mathbb{N}$ for $1 \leq i \leq m$, satisfying

$$j^m(j_1^m(z), \ldots, j_m^m(z)) = z$$

for all $z \in \mathbb{N}$, and given by:

$$\begin{array}{rcl} j_1^1(z) & = & z \\ j_i^{m+1}(z) & = & \left\{ \begin{array}{rl} j_i^m(j_1(z)) & \text{if } 1 \leq i \leq m \\ j_2(z) & \text{if } i = m+1 \end{array} \right. \end{array}$$

Exercise 35 Prove:

i) $j_i^m(j^m(x_1, \ldots, x_m)) = x_i$ for $1 \leq i \leq m$; and

ii) the functions j^m and j_i^m are primitive recursive.

Exercise 35 states that for every m and i, the function j_i^m is primitive recursive. However, the functions j_i^m are connected in such a way, that one is led to suppose that there is also one big primitive recursive function which takes m and i as variables. This is articulated more precisely in the following proposition.

Proposition 3.5 *The function F, defined by*

$$F(x, y, z) = \left\{ \begin{array}{rl} 0 & \textit{if } y = 0 \textit{ or } y > x \\ j_y^x(z) & \textit{else} \end{array} \right.$$

is primitive recursive.

Proof. We first note that the function $G(w, z) = (j_1)^w(z)$ (the function j_1 iterated w times) is primitive recursive. Indeed: $G(0, z) = z$ and $G(w + 1, z) = j_1(G(w, z))$. Now we have:

$$F(x, y, z) = \left\{ \begin{array}{rl} 0 & \text{als } y = 0 \text{ of } y > x \\ G(x - 1, z) & \text{als } y = 1 \\ j_2(G(x - y, z)) & \text{als } y > 1 \end{array} \right.$$

Hence F is defined from the primitive recursive function G by means of repeated distinction by cases. ∎

Exercise 36 Fill in the details of this proof. That is, show that the given definition of F is correct, and that from this definition it follows that F is a primitive recursive function

The functions j^m and their projections j_i^m give primitive recursive bijections: $\mathbb{N}^m \to \mathbb{N}$. Using proposition 3.5 we can now define a bijection: $\bigcup_{m\geq 0}\mathbb{N}^m \to \mathbb{N}$ with good properties. An element of $\mathbb{N}^m$ for $m \geq 1$ is an ordered m-tuple or *sequence* $(x_1, \ldots, x_m)$ of elements of $\mathbb{N}$; the unique element of $\mathbb{N}^0$ is the *empty sequence* $(-)$. The result of the function $\bigcup_{m\geq 0}\mathbb{N}^m \to \mathbb{N}$ to be defined, on input $(x_1, \ldots, x_m)$ or $(-)$ will be written as $\langle x_1, \ldots, x_m\rangle$ or $\langle\,\rangle$ and will be called the *code of the sequence.*

Definition 3.6

$$\begin{array}{rcl} \langle\,\rangle & = & 0 \\ \langle x_0, \ldots, x_{m-1}\rangle & = & j(m-1, j^m(x_0, \ldots, x_{m-1})) + 1 \text{ if } m > 0 \end{array}$$

Exercise 37 Prove that for every $y \in \mathbb{N}$ the following holds: either $y = 0$ or there is a unique $m > 0$ and a unique sequence $(x_0, \ldots, x_{m-1})$ such that $y = \langle x_0, \ldots, x_{m-1}\rangle$.

Remark. In coding arbitrary sequences we have started the convention of letting the indices run from 0; this is more convenient and also consistent with the convention that the natural numbers start at 0.

We now need a few primitive recursive functions for the effective manipulation of sequences.

Definition 3.7 The function $\mathrm{lh}(x)$ gives us the *length* of the sequence with code x, and is given as follows:

$$\mathrm{lh}(x) = \begin{cases} 0 & \text{if } x = 0 \\ j_1(x-1)+1 & \text{if } x > 0 \end{cases}$$

The functions $(x)_i$ give us the i-th element of the sequence with code x (count from 0) if $0 \leq i < \mathrm{lh}(x)$, and 0 otherwise, and is given by

$$(x)_i = \begin{cases} j_{i+1}^{\mathrm{lh}(x)}(j_2(x-1)) & \text{if } x > 0 \text{ and } 0 \leq i < \mathrm{lh}(x) \\ 0 & \text{else} \end{cases}$$

Exercise 38 Prove that the functions $\lambda x.\mathrm{lh}(x)$ and $\lambda xi.(x)_i$ are primitive recursive;
Show that $(\langle x_0, \ldots, x_{m-1}\rangle)_i = x_i$ and that $(\langle\,\rangle)_i = 0$;
Show that for all x: either $x = 0$ or $x = \langle (x)_0, \ldots, (x)_{\mathrm{lh}(x)-1}\rangle$.

The *concatenation function* gives for each x and y the code of the sequence which we obtain by putting the sequences coded by x and y after each other, and is written $x \star y$. That means:

$$\begin{array}{rcl} \langle\,\rangle \star y & = & y \\ x \star \langle\,\rangle & = & x \\ \langle (x)_0, \ldots, (x)_{\mathrm{lh}(x)-1} \rangle \star \langle (y)_0, \ldots (y)_{\mathrm{lh}(y)-1} \rangle & = & \langle (x)_0, \ldots (x)_{\mathrm{lh}(x)-1}, (y)_0, \\ & & \ldots, (y)_{\mathrm{lh}(y)-1} \rangle \end{array}$$

Exercise 39 Show that $\lambda xy.x \star y$ primitive recursive. (Hint: you can first define a primitive recursive function $\lambda xy.x \circ y$, satisfying

$$x \circ y = x \star \langle y \rangle$$

Then, define by primitive recursion a function $F(x, y, w)$ by putting

$$\begin{array}{rcl} F(x, y, 0) & = & x \\ F(x, y, w+1) & = & F(x, y, w) \circ (y)_w \end{array}$$

Finally, put $x \star y = F(x, y, \mathrm{lh}(y))$.)

Course-of-values recursion The scheme of primitive recursion:

$$F(y+1, \vec{x}) = H(y, F(y, \vec{x}), \vec{x})$$

allows us to define the value of $F(y+1, \vec{x})$ directly in terms of $F(y, \vec{x})$. Course-of-values recursion is a scheme which defines $F(y+1, \vec{x})$ in terms of *all* previous values $F(0, \vec{x}), \ldots, F(y, \vec{x})$.

Definition 3.8 Let $G : \mathbb{N}^k \to \mathbb{N}$ and $H : \mathbb{N}^{k+2} \to \mathbb{N}$ be functions. De function $F : \mathbb{N}^{k+1} \to \mathbb{N}$, defined by the clauses

$$\begin{array}{rcl} F(0, \vec{x}) & = & G(\vec{x}) \\ F(y+1, \vec{x}) & = & H(y, \tilde{F}(y, \vec{x}), \vec{x}) \\ (\text{where } \tilde{F}(y, \vec{x}) & = & j^{y+1}(F(0, \vec{x}), \ldots, F(y, \vec{x})) \end{array}$$

is said to be defined from G and H by *course-of-values recursion.*

Proposition 3.9 *Suppose $G : \mathbb{N}^k \to \mathbb{N}$ and $H : \mathbb{N}^{k+2} \to \mathbb{N}$ are primitive recursive functions and $F : \mathbb{N}^{k+1} \to \mathbb{N}$ is defined from G and H by course-of-values recursion. Then F is primitive recursive.*

Proof. Define the function F' as follows:

$$\begin{array}{rcl} F'(0,\vec{x}) & = & G(\vec{x}) \\ F'(y+1,\vec{x}) & = & j(F'(y,\vec{x}),H(y,F'(y,\vec{x}),\vec{x})) \end{array}$$

Then clearly the function F' is primitive recursive. Now we get by induction on y that $F'(y,\vec{x}) = \tilde{F}(y,\vec{x})$:
$F'(0,\vec{x}) = G(\vec{x}) = j^1(G(\vec{x})) = j^1(F(0,\vec{x})) = \tilde{F}(0,\vec{x})$
$F'(y+1,\vec{x}) = j(F'(y,\vec{x}),H(y,F'(y,\vec{x}),\vec{x})) =$
(by induction hypothesis, used twice)
$= j(j^{y+1}(F(0,\vec{x}),\ldots,F(y,\vec{x})),F(y+1,\vec{x})) =$
$= j^{y+2}(F(0,\vec{x}),\ldots,F(y+1,\vec{x}))$
$= \tilde{F}(y+1,\vec{x})$.

We conclude: $F(y,\vec{x}) = \left\{ \begin{array}{rl} G(\vec{x}) & \text{if } y=0 \\ j_2(F'(y,\vec{x})) & \text{else} \end{array} \right.$,
hence $F(y,\vec{x}) = C(G(\vec{x}),j_2(F'(y,\vec{x})),y)$ where C is the function from exercise 28. Therefore F is primitive recursive. ■

We might also consider the following generalization of the course-of-values recursion scheme: instead of allowing only the values $F(w,\vec{x})$ for $w \leq y$ to be used in the definition of $F(y+1,\vec{x})$, we could allow all values $F(w,\vec{x'})$ (for $w \leq y$). This should be well-defined, for inductively we have already defined all functions $F_w = \lambda\vec{x}.F(w,\vec{x})$ when we are defining F_{y+1}. That this is indeed possible (and does not lead us outside the class of primitive recursive functions) if $\vec{x'}$ is a primitive recursive function of $\vec{x}$, is shown in the following exercise.

Exercise 40 Let $K : \mathbb{N} \to \mathbb{N}$, $G : \mathbb{N}^{k+1} \to \mathbb{N}$ and $H : \mathbb{N}^{k+3} \to \mathbb{N}$ be functions. Define F by:

$$\begin{array}{rcl} F(0,\vec{y},x) & = & G(\vec{y},x) \\ F(z+1,\vec{y},x) & = & H(z,F(z,\vec{y},K(x)),\vec{y},x) \end{array}$$

Suppose that G, H and K are primitive recursive.

a) Prove directly, using the pairing function j and suitably adapting the proof of proposition 3.9: if $\forall x(K(x) \leq x)$, then F is primitive recursive.

b) Define a new function F' by:

$$\begin{array}{rcl} F'(0,m,\vec{y},x) & = & G(\vec{y},K^m(x)) \\ F'(n+1,m,\vec{y},x) & = & H(n,F'(n,m,\vec{y},x),\vec{y},K^{m\dot{-}(n+1)}(x)) \end{array}$$

Prove: if $n \leq m$ then $\forall k[F'(n, m+k, \vec{y}, x) = F'(n, m, \vec{y}, K^k(x))]$

c) Prove by induction: $F(z, \vec{y}, x) = F'(z, z, \vec{y}, x)$ and conclude that F is primitive recursive, also without the assumption that $K(x) \leq x$.

Double recursion. However, the matter is totally different if, in the definition of $F(y+1, \vec{x})$, we allow values of F_y at arguments in which already known values of F_{y+1} may appear. In this case we speak of *double recursion.* We treat a simple case, with a limited number of variables.

Definition 3.10 Let $G : \mathbb{N} \to \mathbb{N}$, $H : \mathbb{N}^2 \to \mathbb{N}$, $K : \mathbb{N}^4 \to \mathbb{N}$, $J : \mathbb{N} \to \mathbb{N}$, en $L : \mathbb{N}^3 \to \mathbb{N}$ be functions; the function F is said to be defined from these by *double recursion* if

$$\begin{array}{rcl} F(0,z) & = & G(z) \\ F(y+1,0) & = & H(y, F(y, J(y))) \\ F(y+1,z+1) & = & K(y, z, F(y+1,z), F(y, L(y,z,F(y+1,z)))) \end{array}$$

Proposition 3.11 *If G, H, K, J and L are primitive recursive and F is defined from these by double recursion as in definition 3.10 then all functions $F_y = \lambda z.F(y,z)$ are primitive recursive, but F itself need not be primitive recursive.*

Proof. It follows from the definition that all functions F_y are primitive recursive. We give an example of a non-primitive recursive function that can be defined by double recursion. The idea is, to code all definitions of primitive recursive functions $\mathbb{N} \to \mathbb{N}$ as numbers, in the following way:

- The basic functions are the functions $\lambda x.0$, $\lambda x.x+1$ and j_i^m, which get codes $\langle 0 \rangle$, $\langle 1 \rangle$ and $\langle 2, i, m \rangle$ respectively;

- if $H, G_1, \ldots, G_p$ have codes $n, m_1, \ldots, m_p$ respectively, and F is defined by
$$F(x) = H(j^p(G_1(x), \ldots, G_p(x)))$$
then F has code $\langle 3, n, m_1, \ldots, m_p \rangle$;

- if H and G have codes n and m and F is defined by
$$\begin{array}{rcl} F(j(x,0)) & = & G(x) \\ F(j(x,y+1)) & = & H(j^3(x, F(j(x,y)), y)) \end{array}$$
then F has code $\langle 4, n, m \rangle$.

Check for yourself that every primitive recursive function of one variable can be defined by the clauses above, and hence has a code (actually, more than one, because there are many definitions of one and the same primitive recursive function).

The next step in the proof is now to define a function Val (actually by double course-of-value recursion) of two variables k and n, such that the following holds: if k is the code of a definition of a primitive recursive function F, then $\mathrm{Val}(k,n) = F(n)$. This is done as follows:

$$\mathrm{Val}(k,x) = \begin{cases} \hfill 0 \text{ if } k = \langle 0 \rangle \\ \hfill x+1 \text{ if } k = \langle 1 \rangle \\ \hfill j_i^m(x) \text{ if } k = \langle 2, i, m \rangle \\ \hfill \mathrm{Val}(n, j^p(\mathrm{Val}(m_1,x),\ldots,\mathrm{Val}(m_p,x))) \\ \hfill \text{if } k = \langle 3, n, m_1, \ldots, m_p \rangle \\ \hfill \mathrm{Val}(m, j_1(x)) \text{ if } k = \langle 4, n, m \rangle \text{ and } j_2(x) = 0 \\ \hfill \mathrm{Val}(n, j^3(j_1(x), \mathrm{Val}(k, j(j_1(x), j_2(x)-1)), j_2(x)-1)) \\ \hfill \text{if } k = \langle 4, n, m \rangle \text{ and } j_2(x) > 0 \\ \hfill 0 \text{ else} \end{cases}$$

Note that $\mathrm{Val}(k,x)$ is defined in terms of $\mathrm{Val}(n,y)$ for $n < k$ or $n = k$ and $y < x$; so Val is well-defined as a function.

The apotheosis of the proof is an example of *diagonalisation*, a form of reasoning similar to Cantor's proof of the uncountability of the set of real numbers; this is a technique we shall meet more often.

Suppose the function Val is primitive recursive. Then so is the function $\lambda x.\mathrm{Val}(x,x)+1$, which is a function of one variable; this function has therefore a code, say k.
But now by construction of Val, we have that $\mathrm{Val}(k,k) = \mathrm{Val}(k,k)+1$; which is a contradiction. We conclude that the function Val, which was defined by double recursion from primitive recursive functions, is not primitive recursive, which is what we set out to show. ■

Comparing the definition schemes of primitive recursion and double recursion, we see that in the first case the function $F_y = \lambda x.F(y,x)$ is applied to the argument x, whereas in the second case F_y is applied to arguments which may contain values of F_{y+1}. This allows functions defined by double recursion to "grow very fast".

In 1927, the Romanian mathematician Sudan ([24]) gave an example of a "computable" function (a function, for which an algorithm exists to compute it) which is not primitive recursive. In 1928, W. Ackermann ([1]) gave an example of a function $G(x,y)$ of two variables, defined by double recursion from primitive recursive functions, which has the following property: for every unary primitive recursive function $F(x)$ there is a number x_0 such that for all $x > x_0$, $F(x) < G(x, x)$. Check yourself that it follows that G cannot be primitive recursive! Such functions G are called *Ackermann functions.*

Ackermann's example was later simplified by Rosza Péter; this simplifications is presented in the exercise below.

Exercise 41 (Ackermann-Péter) Define by double recursion:

$$\begin{array}{rcl} A(0,x) & = & x+1 \\ A(n+1,0) & = & A(n,1) \\ A(n+1,x+1) & = & A(n,A(n+1,x)) \end{array}$$

Again we write A_n for $\lambda x.A(n,x)$. For a primitive recursive function $F : \mathbb{N}^k \to \mathbb{N}$, we say that F is *bounded by* A_n, written $F \in \mathcal{B}(A_n)$, if for all $x_1, \ldots, x_k$ we have $F(x_1, \ldots, x_k) < A_n(x_1 + \cdots + x_k)$. Prove by inductions on n and x:

i) $n + x < A_n(x)$

ii) $A_n(x) < A_n(x+1)$

iii) $A_n(x) < A_{n+1}(x)$

iv) $A_n(A_{n+1}(x)) \leq A_{n+2}(x)$

v) $nx + 2 \leq A_n(x)$ for $n \geq 1$

vi) $\lambda x.x+1$, $\lambda x.0$ and $\lambda \vec{x}.x_i \ \in \mathcal{B}(A_1)$

vii) if $F = \lambda \vec{x}.H(G_1(\vec{x}), \ldots, G_p(\vec{x}))$ and $H, G_1, \ldots, G_p \ \in \mathcal{B}(A_n)$ for some $n > p$, then $F \in \mathcal{B}(A_{n+2})$

viii) for every $n \geq 1$ we have: if $F(0,\vec{x}) = G(\vec{x})$ and $F(y+1,\vec{x}) = H(y, F(y,\vec{x}), \vec{x})$ and $G, H \ \in \mathcal{B}(A_n)$, then $F \in \mathcal{B}(A_{n+3})$

Conclude that for every primitive recursive function F there is a number n such that $F \in \mathcal{B}(A_n)$; hence, that A is an Ackermann function.

Exercise 42 Define a sequence of functions $G_0, G_1, \ldots : \mathbb{N} \to \mathbb{N}$ by

$$\begin{array}{rcl} G_0(y) & = & y+1 \\ G_{x+1}(y) & = & (G_x)^{y+1}(y) \end{array}$$

and then define G by putting $G(x,y) = G_x(y)$. Give a definition of G by double rcursion and composition (use a definition scheme for double recursion which allows an extra variable) and prove that G is an Ackermann function.

A few simple exercises to conclude this section:

Exercise 43 Show that the following "recursion scheme" does not define a function:

$$\begin{array}{rcl} F(0,0) & = & 0 \\ F(x+1,y) & = & F(y,x+1) \\ F(x,y+1) & = & F(x+1,y) \end{array}$$

Exercise 44 Show that the following "recursion scheme" is not satisfied by any function:

$$\begin{array}{rcl} F(0,0) & = & 0 \\ F(x+1,y) & = & F(x,y+1)+1 \\ F(x,y+1) & = & F(x+1,y)+1 \end{array}$$

3.3 Partial recursive functions

Definition 3.12 Let X and Y be sets. A *partial function* F from X to Y is a function $F : U \to Y$ where U is a subset of X. We call U the *domain of* F, and write $\mathrm{dom}(F)$. We write $F : X \rightharpoonup Y$ to indicate that F is a partial function from X to Y. The function F is *total* if $\mathrm{dom}(F) = X$. We treat total functions as a special case of partial functions; a partial function *may* be total.

If $x \in X$, we say "$F(x)$ is defined" if $x \in \mathrm{dom}(F)$.

Partial functions can be composed: if $F : X \rightharpoonup Y$ and $G : Y \rightharpoonup Z$ then $GF : X \rightharpoonup Z$ is the function whose domain is the subset $\{x \in X \mid x \in \mathrm{dom}(F) \text{ and } F(x) \in \mathrm{dom}(G)\}$ of X.

We shall use the symbol $\simeq$ (*Kleene equality*) between expressions $F(x)$ and $G(x)$ for partial functions: $F(x) \simeq G(x)$ means that $F(x)$ is defined precisely when $G(x)$ is defined, and whenever this is the case, $F(x) = G(x)$. In particular, $F(x) \simeq G(x)$ holds if both sides are undefined.

Composite terms built up from partial functions are interpreted in the way we have defined composition. That means, that a term cannot be defined unless all its subterms are defined. Example: if Π^2_1 denotes the first projection $\mathbb{N}^2 \to \mathbb{N}$ as before, and $G : \mathbb{N} \rightharpoonup \mathbb{N}$ is a partial function, then $\Pi^2_1(x, G(y))$ is only defined when $G(y)$ is defined, and $\Pi^2_1(x, G(y)) \simeq x$ need not hold.

Definition 3.13 The class of *partial recursive functions* $\mathbb{N}^k \rightharpoonup \mathbb{N}$ (for variable k) is generated by the following clauses:

i) all primitive recursive functions are partial recursive;

ii) the partial recursive functions are closed under composition: if $G_1, \ldots, G_l : \mathbb{N}^k \rightharpoonup \mathbb{N}$ en $H : \mathbb{N}^l \rightharpoonup \mathbb{N}$ are partial recursive, then so is the function $\lambda\vec{x}.H(G_1(\vec{x}), \ldots, G_l(\vec{x}))$. This function is defined for all $\vec{x} \in \bigcap_{i=1}^{l} \mathrm{dom}(G_i)$ for which
$(G_1(\vec{x}), \ldots, G_l(\vec{x})) \in \mathrm{dom}(H)$;

iii) if $G : \mathbb{N}^{k+1} \rightharpoonup \mathbb{N}$ is partial recursive, then also F when F is defined from G by *minimization*: we write

$$F(\vec{x}) \simeq \mu y.G(\vec{x}, y) = 0$$

$F(\vec{x})$ is defined precisely when there exists a y such that $\forall i \leq y.(\vec{x}, i) \in \mathrm{dom}(G)$ and $G(\vec{x}, y) = 0$. $F(\vec{x})$ then denotes the least y with that property.

Definition 3.14 A relation $A \subseteq \mathbb{N}^k$ is called recursive if its characteristic function χ_A is partial recursive.

A partial recursive function is *total recursive* or *recursive* if it is total. Because χ_A is always a total function for every relation A, there is no notion of “partial recursive relation”.

Proposition 3.15

i) If R is a $k+1$-ary recursive relation and $F : \mathbb{N}^k \rightharpoonup \mathbb{N}$ is defined by

$$F(\vec{x}) \simeq \mu y.R(\vec{x}, y)$$

then F is partial recursive;

ii) *If R is a recursive relation and G is a partial recursive function, and F is defined by*

$$F(x) \simeq \begin{cases} G(x) \ \textit{if} \ \exists y.R(y,x) \\ \textit{undefined else} \end{cases}$$

then F is partial recursive;

Proof. For,

i) $F(\vec{x}) \simeq \mu y.\chi_R(\vec{x}, y) = 0$

ii) $F(x) \simeq G((\mu y.\chi_R(y,x) = 0)0 + x)$. Recall our convention about when terms are defined!

■

3.4 *Smn*-Theorem and Recursion Theorem

In the 1930's, logicians were concerned with the question what a "computable" function should be: a (possibly partial) function $F : \mathbb{N}^k \rightharpoonup \mathbb{N}$ for which there exists an algorithm that allows a person (or a computer) to calculate, step by step, the value of F at given arguments. There may be arguments for which the algorithm, when carried out, never reaches a final state.

Now what is an algorithm? Before we can think further about computable functions, we should have a clear notion of this. It turned out that several competing definitions of the notion of "algorithm" (advanced by Alonzo Church, Stephen Cole Kleene and Alan Turing) yielded the *same* notion of partial computable function: namely, partial recursive function.

The notion of a partial recursive function is therefore a very natural one, and is studied in the area of Logic called *Recursion Theory* or *Computability Theory.* In a course in Recursion Theory, you will learn about the equivalence between partial recursive and algorithmically computable. In this course, we don't have time for such a treatment, and therefore we state some theorems without proof.

Theorem 3.16 (Kleene Enumeration Theorem) *There is a quaternary (4-ary) primitive recursive relation T and a unary primitive recursive function U such that for every partial recursive function $F : \mathbb{N}^k \rightharpoonup \mathbb{N}$ there exists a number e (the* index *of the function F) with the following properties:*

i) *For all k-tuples $n_1, \ldots, n_k$ we have: $F(n_1, \ldots, n_k)$ is defined precisely when there is a number y such that $T(k, e, j^k(n_1, \ldots, n_k), y)$ holds (that is, $(k, e, j^k(n_1, \ldots, n_k), y) \in T$);*

ii) *If $F(n_1, \ldots, n_k)$ is defined then $F(n_1, \ldots, n_k) = U(y)$ for the least y as in i).*

If e corresponds to the k-ary partial recursive function F as in Theorem 3.16 we have therefore:

$$F(\vec{n}) \;\simeq\; U(\mu y.T(k, e, j^k(\vec{n}), y))$$

and we write $\varphi_e^{(k)}$ for F.

The letters T and U are standard in Computability Theory. The relation T is also called the *Kleene T-predicate* (*predicate* is another word for relation) and U is the *result extraction function.*

Since the relation T is primitive recursive, the partial function

$$\Psi(m, e, x) \;\simeq\; U(\mu y.T(m, e, x, y))$$

is partial recursive, and *every* k-ary partial recursive function is of the form $\lambda x_1 \cdots x_k.\Psi(k, e, j^k(x_1, \ldots, x_k))$ for some e. An algorithm for the function Ψ is therefore called a *universal* algorithm.

In contrast with this, we do *not* have a "universal algorithm" for *total* recursive functions:

Proposition 3.17 *There is no total recursive function $\Psi(m, e, x)$ such that every total recursive function $F : \mathbb{N}^m \to \mathbb{N}$ equals*

$$\lambda x_1 \cdots x_m.\Psi(m, e, j^m(x_1, \ldots, x_m))$$

for a certain e.

Proof. For suppose to the contrary that such a function Ψ exists. Then the function

$$\lambda x_1 \cdots x_m.\Psi(m, j^m(x_1, \ldots, x_m), j^m(x_1, \ldots, x_m)) + 1$$

is total recursive, hence equal to $\lambda x_1 \cdots x_m.\Psi(m, e, j^m(x_1, \ldots, x_m))$ for a certain e; but for that e we would have

$$\Psi(m, e, e) = \Psi(m, e, e) + 1$$

and we obtain a contradiction (note that this is a diagonalisation similar to the proof of 3.11). ■

The following important theorem has the funny (and non-descriptive) name of "*Smn* Theorem". A better name would be "Parametrization Theorem", because it says that indices of partial recursive functions with parameters can be obtained primitive-recursively in the parameters. We do not go into the proof.

Theorem 3.18 (*Smn*-Theorem; Kleene) *For every $m \geq 1$ and $n \geq 1$ there is an $m+1$-ary primitive recursive function S^m_n such that for all $e, x_1, \ldots, x_m, y_1, \ldots, y_n$ we have:*

$$\varphi^{(n)}_{S^m_n(e,x_1,\ldots,x_m)}(y_1,\ldots,y_n) \simeq \varphi^{m+n}_e(x_1,\ldots,x_m,y_1,\ldots,y_n)$$

Corollary 3.19 *There is a primitive recursive function H such that for all e, f, x we have:*

$$\varphi^{(1)}_{H(e,f)}(x) \simeq \varphi^{(1)}_e(\varphi^{(1)}_f(x))$$

Proof. The function $\lambda efx.\varphi^{(1)}_e(\varphi^{(1)}_f(x))$ is partial recursive; for

$$\varphi^{(1)}_e(\varphi^{(1)}_f(x)) \simeq U(j_2(\mu z.[T(1,f,x,j_1(z)) \wedge T(1,e,U(j_1(z)),j_2(z))]))$$

Hence $\varphi^{(1)}_e(\varphi^{(1)}_f(x)) \simeq \varphi^{(3)}_g(e,f,x)$ for a certain index g; put $H(e,f) = S^2_1(g,e,f)$ ■

Our next consequence of the *Smn*-Theorem looks rather bizarre at first sight. It allows us to find an index for a partial recursive function, satisfying a property which depends on the index we want to find!

Corollary 3.20 (Recursion Theorem, Kleene 1938) *For every partial recursive function $F : \mathbb{N}^{k+1} \rightharpoonup \mathbb{N}$ with $k \geq 1$ there is an index e such that for all $x_1, \ldots, x_k$ the following holds:*

$$\varphi^{(k)}_e(x_1,\ldots x_k) \simeq F(x_1,\ldots,x_k,e)$$

Proof. Let f be an index for F, so $\varphi^{(k+1)}_f(x_1,\ldots,x_{k+1}) \simeq F(x_1,\ldots,x_{k+1})$ voor alle $x_1,\ldots,x_{k+1}$. Now let g be an index which satisfies, for all $h, y, x_1, \ldots, x_k$:

$$\varphi^{(k+2)}_g(h,y,x_1,\ldots,x_k) \simeq \varphi^{(k+1)}_h(x_1,\ldots,x_k,S^1_k(y,y))$$

(Note that the expression on the RHS is a partial recursive function of $h, y, x_1, \ldots, x_k$, so such an index g exists)

Now define

$$e = S^1_k(S^1_{k+1}(g,f), S^1_{k+1}(g,f))$$

Then we have:

$$\begin{array}{rcl}
\varphi^{(k)}_e(x_1,\ldots,x_k) & \simeq & \\
\varphi^{(k)}_{S^1_k(S^1_{k+1}(g,f),S^1_{k+1}(g,f))}(x_1,\ldots,x_k) & \simeq & \text{by the } Smn\text{-Theorem} \\
\varphi^{(k+1)}_{S^1_{k+1}(g,f)}(S^1_{k+1}(g,f),x_1,\ldots,x_k) & \simeq & \\
\varphi^{(k+2)}_g(f,S^1_{k+1}(g,f),x_1,\ldots,x_k) & \simeq & \text{by choice of } g \\
\varphi^{(k+1)}_f(x_1,\ldots,x_k,S^1_k(S^1_{k+1}(g,f),S^1_{k+1}(g,f))) & \simeq & \text{by definition of } e \\
\varphi^{(k+1)}_f(x_1,\ldots,x_k,e) & \simeq & \text{by choice of } f \\
F(x_1,\ldots,x_k,e) & &
\end{array}$$

■

Exercise 45 Let $R_1,\ldots,R_n \subseteq \mathbb{N}^k$ be recursive relations such that $R_i \cap R_j = \emptyset$ for $i \neq j$; suppose $G_1,\ldots,G_n : \mathbb{N}^k \rightharpoonup \mathbb{N}$ are partial recursive. Then the partial function F, defined by

$$F(\vec{x}) \simeq \begin{cases} G_1(\vec{x}) & \text{if } R_1(\vec{x}) \\ \vdots & \vdots \\ G_n(\vec{x}) & \text{if } R_n(\vec{x}) \\ \text{ongedefinieerd} & \text{anders} \end{cases}$$

is also partial recursive. Prove this.

Corollary 3.21 *The class of partial recursive functions is closed under primitive recursion: if $G : \mathbb{N}^k \rightharpoonup \mathbb{N}$ and $H : \mathbb{N}^{k+2} \rightharpoonup \mathbb{N}$ are partial recursive, and F is defined from G and H by primitive recursion, then F is also partial recursive.*

Proof. Let g and h be indices for G and H, respectively. By the Recursion Theorem there is an index f such that for all $y, \vec{x}$:

$$\varphi^{(k+1)}_f(y,\vec{x}) = \begin{cases} \varphi^{(k)}_g(\vec{x}) & \text{if } y = 0 \\ \varphi^{(k+2)}_h(y-1, \varphi^{(k+1)}_f(y-1,\vec{x}),\vec{x}) & \text{if } y > 0 \end{cases}$$

Check yourself that $\varphi^{(k+1)}_f(y,\vec{x}) \simeq F(y,\vec{x})$ voor alle $y, \vec{x}$. ■

Corollary 3.22 *The recursive relations are closed under bounded quantifiers: if $R \subseteq \mathbb{N}^{k+1}$ is recursive, then*

$$\{(\vec{x}, y) \mid \forall w < y . R(\vec{x}, w)\}$$

and

$$\{(\vec{x}, y) \mid \exists w < y . R(\vec{x}, w)\}$$

are also recursive.

Proof. For, the characteristic functions of these relations are defined by primitive recursion from the characteristic function of R. ■

Having another look at the proof of the Recursion Theorem, we see that the index e we found there, is actually the result of a primitive recursive function applied to an index f of F. In other words:

> There is a primitive recursive function G_n such that for all $x_1, \ldots, x_n, f$:
>
> $$\varphi^{(n)}_{G_n(f)}(x_1, \ldots, x_n) \simeq \varphi^{(n+1)}_f(x_1, \ldots, x_n, G_n(f))$$

Exercise 46 Show that the following indices for partial recursive functions can be found:

i) Given a recursive relation R, find e such that for all $\vec{x}$:

$$\varphi^{(k)}_e(\vec{x}) = \begin{cases} 0 & \text{if } R(\vec{x}, e) \\ 1 & \text{else} \end{cases}$$

ii) Given a recursive relation R and a partial recursive function F, find e such that for all $\vec{x}$:

$$\varphi^{(k)}_e(\vec{x}) = \begin{cases} F(\vec{x}) & \text{if } \exists y . R(\vec{x}, y, e) \\ \text{undefined} & \text{else} \end{cases}$$

Exercise 47 Prove the *Recursion Theorem in parameters*: there is a primitive recursive function F such that for all $f, y_1, \ldots, y_n, x_1, \ldots, x_m$:

$$\varphi^{(m)}_{F(f,\vec{y})}(\vec{x}) \simeq \varphi^{(n+1)}_f(F(f, \vec{y}), \vec{y}, \vec{x})$$

and also: there is a primitive recursive F' such that for all $f, y_1, \ldots, y_n$ and $x_1, \ldots, x_m$:

$$\varphi^{(m)}_{F'(f,\vec{y})}(\vec{x}) \simeq \varphi^{(m)}_{\varphi^{(n+1)}_f(F'(f,\vec{y}),\vec{y})}(\vec{x})$$

Concluding this chapter, let us prove that the class of total recursive functions is closed under double recursion. Suppose therefore that G, H, J, K and L are total recursive, and let F be defined by:

$$\begin{array}{rcl} F(0,z) & = & G(z) \\ F(y+1,0) & = & H(y,F(y,J(y))) \\ F(y+1,z+1) & = & K(y,z,F(y+1,z),F(y,L(y,z,F(y+1,z)))) \end{array}$$

Then F is also total recursive; for we can use the Recursion Theorem in order to find an index f such that

$$\varphi^{(2)}_f(y,z) = \left\{ \begin{array}{r} G(z) \text{ if } y=0 \\ H(y{-}1,\varphi^{(2)}_f(y{-}1,J(y{-}1))) \text{ if } y>0 \text{ and } z=0 \\ K(y{-}1,z{-}1,\varphi^{(2)}_f(y,z{-}1),\varphi^{(2)}_f(y{-}1,L(y{-}1,z{-}1,\varphi^{(2)}_f(y,z{-}1)))) \\ \text{if } y>0 \text{ and } z>0 \end{array} \right.$$

Exercise 48 Prove by double induction (on y and z) that the function φ_f defined above, is total and equal to F.

One last exercise.

Exercise 49 Prove *Smullyan's Double Recursion Theorem*: given two 2-ary partial recursive functions F and G, for every k there are indices a and b such that for all $x_1,\ldots,x_k$:

$$\varphi^{(k)}_a(x_1,\ldots,x_k) \simeq \varphi^{(k)}_{F(a,b)}(x_1,\ldots,x_k)$$

en

$$\varphi^{(k)}_b(x_1,\ldots,x_k) \simeq \varphi^{(k)}_{G(a,b)}(x_1,\ldots,x_k)$$

For further reference, a definition, and a theorem without proof.

Definition 3.23 A subset $A \subseteq \mathbb{N}^k$ is called *recursively enumerable* or *r.e.* if there is a primitive recursive subset $U \subseteq \mathbb{N}^{k+1}$ such that

$$A \;=\; \{\vec{x} \in \mathbb{N}^k \,|\, \exists y.(y,\vec{x}) \in U)\}$$

Exercise 50 Show that a subset $A \subseteq \mathbb{N}^k$ is recursively enumerable if and only if there is a k-ary partial recursive function F such that $A = \mathrm{dom}(F)$.

Clearly, every recursive set is r.e.

Exercise 51 Prove, that a set $A \subseteq \mathbb{N}^k$ is recursive, precisely when A and $\mathbb{N}^k - A$ are both recursively enumerable.

Theorem 3.24 (Turing) *The set*

$$\{(e, \vec{x}) \in \mathbb{N}^{k+1} \,|\, \vec{x} \in \mathrm{dom}(\varphi_e^{(k)})\}$$

is recursively enumerable, but not recursive.

Chapter 4

The Formal System of Peano Arithmetic

The system of first-order *Peano Arithmetic* or PA, is a theory in the language $\mathcal{L}_{\mathrm{PA}} = \{0, 1; +, \cdot\}$ where $0, 1$ are constants, and $+, \cdot$ binary function symbols. It has the following axioms:

1) $\forall x \neg(x + 1 = 0)$

2) $\forall xy(x + 1 = y + 1 \to x = y)$

3) $\forall x(x + 0 = x)$

4) $\forall xy(x + (y + 1) = (x + y) + 1)$

5) $\forall x(x{\cdot}0 = 0)$

6) $\forall xy(x{\cdot}(y + 1) = (x{\cdot}y) + x)$

7) $\forall \vec{x}[(\varphi(0, \vec{x}) \wedge \forall y(\varphi(y, \vec{x}) \to \varphi(y + 1, \vec{x}))) \to \forall y \varphi(y, \vec{x})]$

Item 7 is meant to be an axiom for every formula $\varphi(y, \vec{x})$. These axioms are called *induction axioms*. Such a set of axioms, given by one or more generic symbols "φ" which range over all formulas, is called an *axiom scheme*; in our case we talk about the *induction scheme*.

So, PA is given by infinitely many axioms and this infinitude is essential: there is no finite $\mathcal{L}_{\mathrm{PA}}$-theory which has the same models as PA.

Clearly, the set $\mathbb{N}$ together with the elements 0,1 and usual addition and multiplication, is a model of PA, which we call the *standard model* and denote by $\mathcal{N}$. It is easy to see that PA has also non-standard models. First define, for every $n \in \mathbb{N}$, a term $\overline{n}$ of $\mathcal{L}_{\mathrm{PA}}$ by recursion: $\overline{0} = 0$ and $\overline{n+1} = \overline{n}{+}1$ (mind you, this is not the identity function! E.g., $\overline{3} = ((0+1)+1)+1$). Terms of the form $\overline{n}$ are called *numerals* and we shall use them a lot later on. Now let c be a new constant, and consider in the language $\mathcal{L}_{\mathrm{PA}} \cup \{c\}$ the set of axioms:

$$\{\text{axioms of PA}\} \cup \{\neg(c = \overline{n}) \,|\, n \in \mathbb{N}\}$$

Since every finite subset of this theory has a straighforward interpretation in $\mathbb{N}$, this is (by the Compactness Theorem, Exercise 20) a consistent set of axioms and has therefore a model $\mathcal{M}$, which has a *nonstandard element* $c^{\mathcal{M}}$.

The theory PA is surprisingly strong: it can represent (in a suitable sense, soon to be made precise) all recursive functions, and most elementary number theory can be carried out in this system. Ironically though, it is exactly this strength that lies at the basis of its being *incomplete* as Gödel was the first to show. Since we wish to arrive at these famous *Incompleteness Theorems*, our first aim is to develop some elementary number theory in PA. Our first proposition establishes basic properties of addition and multiplication.

Proposition 4.1

i) $\mathrm{PA} \vdash \forall x(x = 0 \vee \exists y(x = y+1))$

ii) $\mathrm{PA} \vdash \forall xyz(x + (y+z) = (x+y)+z)$

iii) $\mathrm{PA} \vdash \forall xy(x + y = y + x)$

iv) $\mathrm{PA} \vdash \forall xyz(x + z = y + z \to x = y)$

v) $\mathrm{PA} \vdash \forall xyz(x{\cdot}(y{\cdot}z) = (x{\cdot}y){\cdot}z)$

vi) $\mathrm{PA} \vdash \forall xy(x{\cdot}y = y{\cdot}x)$

vii) $\mathrm{PA} \vdash \forall xyz(x{\cdot}(y+z) = (x{\cdot}y) + (x{\cdot}z))$

viii) $\mathrm{PA} \vdash \forall xyz(\neg(z = 0) \wedge x{\cdot}z = y{\cdot}z \to x = y)$

Proof. All of these are proved using the induction axioms. For i), let $\varphi(x)$ be $x = 0 \vee \exists y(x = y+1)$. Clearly, $\mathrm{PA} \vdash \varphi(0) \wedge \forall y \varphi(y+1)$, so $\mathrm{PA} \vdash \forall x \varphi(x)$.

For ii), use "induction on z" that is, let $\varphi(z)$ be the formula $\forall xy(x + (y + z) = (x+y)+z)$. Then $\mathrm{PA} \vdash \varphi(0)$ by axiom 3, and $\mathrm{PA} \vdash \varphi(z) \to \varphi(z+1)$ by axiom 4, since

$$\varphi(z) \vdash (x + (y+z)) + 1 = x + ((y+z)+1) = x + (y + (z+1))$$

The proof of the other statements is a useful exercise (sometimes, as in iii), you will need to perform a double induction). ∎

Exercise 52 Prove statements iii)-viii) of proposition 4.1.

Proposition 4.2 *Let $\varphi(x,y)$ be the formula $\exists z(x+(z+1)=y)$. Then in* PA*, φ defines a discrete linear order with least element which satisfies the least number principle, i.e.*

i) $\text{PA} \vdash \neg\varphi(x,x)$

ii) $\text{PA} \vdash \varphi(x,y) \wedge \varphi(y,z) \rightarrow \varphi(x,z)$

iii) $\text{PA} \vdash \varphi(x,y) \vee x=y \vee \varphi(y,x)$

iv) $\text{PA} \vdash x=0 \vee \varphi(0,x)$

v) $\text{PA} \vdash \varphi(x,y) \rightarrow (y=x+1 \vee \varphi(x+1,y))$

vi) $\text{PA} \vdash \exists w\psi(w) \rightarrow \exists y(\psi(y) \wedge \forall x(\varphi(x,y) \rightarrow \neg\psi(x)))$

vii) $\text{PA} \vdash \varphi(x,x+1)$

Exercise 53 Prove proposition 4.2

The scheme vi) of proposition 4.2 is called the *least number principle* LNP.

Exercise 54 Prove that LNP is equivalent to the scheme of induction, in the following sense: let PA$'$ be the theory with the first 6 axioms of PA, and the statements of proposition 4.2 as axioms. Then PA and PA$'$ are equivalent theories, in the sense that they have the same models.

The order defined in proposition 4.2 is so important that we introduce a new symbol for it: henceforth we write $x<y$ for $\exists z(x+(z+1)=y)$. We shall also use the abbreviations $\exists x<y$ and $\forall x<y$ for $\exists x(x<y \wedge \ldots)$ and $\forall x(x<y \rightarrow \ldots)$, respectively. We shall write $x \leq y$ for $x=y \vee x<y$, and $x \neq y$ for $\neg(x=y)$. This process of introducing abbreviations will continue throughout; it is absolutely essential if we want to write meaningful formal statements (but, especially later when we shall also introduce function symbols, we shall have to make sure that the properties of the meant functions are provable in PA).

What we do is actually this: we shall successively introduce Skolem functions for uniquely defined elements, and enlarge the theory PA by axioms for these function symbols, in such a way as to obtain *a chain of definitional extensions of* PA in the sense of section 2.3.

Exercise 55 Prove the principle of *well-founded induction*, that is:

$$\mathrm{PA} \vdash \forall w(\forall v < w\psi(v) \to \psi(w)) \to \forall w\psi(w)$$

Exercise 56 Prove:

$$\mathrm{PA} \vdash \forall xy(y \neq 0 \to x \leq x{\cdot}y)$$

4.1 Elementary Number Theory in PA

The starting point for our treatment of elementary number theory in PA is the theorem of *Euclidean division*.

Theorem 4.3 (Division with remainder)

$$\mathrm{PA} \vdash \forall xy(y \neq 0 \to \exists ab(x = a{\cdot}y + b \wedge 0 \leq b < y))$$

Moreover, PA *proves that such a, b are unique.*

Proof. By induction on x. Clearly, $0 = 0{\cdot}y + 0$; if $x = a{\cdot}y + b \wedge 0 \leq b < y$ then by 4.2v), $b + 1 < y \vee b + 1 = y$. If $b + 1 < y$, $x + 1 = a{\cdot}y + (b + 1)$ and if $b + 1 = y$, $x + 1 = (a + 1){\cdot}y + 0$.

For uniqueness, suppose $x = a{\cdot}y + b = a'{\cdot}y + b'$ with $0 \leq b, b' < y$. If $a < a'$ then $a + 1 \leq a'$ hence

$$a'{\cdot}y \geq a{\cdot}y + y > a{\cdot}y + b = x$$

with a contradiction. So $a' \leq a$ and by symmetry, $a = a'$. Then $b = b'$ follows by 4.1iv). ∎

In the notation of theorem 4.3, we call b the *remainder of x on division by y*, and a the *integer part of x divided by y*.

Again, we introduce shorthand notation:

$$\begin{aligned} x|y &\equiv \exists z(x{\cdot}z = y) \\ \mathrm{irred}(x) &\equiv \forall v \leq x(v|x \to v = 1 \vee v = x) \\ \mathrm{prime}(x) &\equiv x > 1 \wedge \forall yz(x|(y{\cdot}z) \to x|y \vee x|z) \end{aligned}$$

Furthermore, since PA $\vdash \forall xy\exists!z((z = 0 \wedge x < y) \vee x = z + y)$, we may introduce a function symbol $-$ to the language, with axiom

$$\forall xy((x < y \wedge x - y = 0) \vee (x = y + (x - y)))$$

I hope the notations are familiar. The notions "irreducible" and "prime" element are from ring theory.

Proposition 4.4

$$\mathrm{PA} \vdash \forall x(x > 1 \rightarrow (\mathrm{irred}(x) \leftrightarrow \mathrm{prime}(x)))$$

Proof. If prime(x) and $v|x$ so $v{\cdot}z = x$ then either $x|v$ whence $v = x$, or $x|z$ whence $v = 1$. So irred(x). Conversely suppose irred(x) and $x > 1$. Let $P(v)$ be the formula

$$\forall yz \leq v(y{\cdot}z \leq v \wedge x|(y{\cdot}z) \rightarrow x|y \vee x|z)$$

We show $\forall w(\forall v < wP(v) \rightarrow P(w))$, so by well-founded induction we may conclude $\forall wP(w)$ which clearly implies prime(x).

So suppose $\forall v < wP(v)$ and $y, z \leq w$ such that $y{\cdot}z \leq w$, $x|(y{\cdot}z)$, $x \nmid y$, $x \nmid z$. Then $y, z > 1$ and using 4.3 we may assume $y < x$ since otherwise replace y by its remainder on division by x. Again using 4.3, let $x = a{\cdot}y + b$ with $0 \leq b < y$. If $b = 0$ then by irreducibility of x, $y = 1 \vee y = x$, a contradiction in both cases. If $b > 0$ we have

$$b{\cdot}z = (x - a{\cdot}y){\cdot}z = x{\cdot}z - a{\cdot}y{\cdot}z$$

so $x|(b{\cdot}z)$, $x \nmid b$, $x \nmid z$ and $b{\cdot}z < y{\cdot}z \leq w$; contradiction with $\forall v < wP(v)$. Therefore $P(w)$, and we are done. ∎

Proposition 4.5 PA $\vdash \forall x(x > 1 \rightarrow \exists v(\mathrm{prime}(v) \wedge v|x))$

Proof. If $x > 1$, since $x|x$ we have $\exists w(w > 1 \wedge w|x)$. By LNP, there is a least such w. The least such w is irreducible, hence prime by proposition 4.4. ∎

Exercise 57 Prove that "PA proves the existence of infinitely many primes", i.e. the statement

$$\forall x\exists y(x < y \wedge \mathrm{prime}(y))$$

[Hint: first prove, by induction in PA, $\forall x\exists y > 0\forall i(1 \leq i \leq x \rightarrow i|y)$. Given such y, consider $y + 1$ and apply proposition 4.5]

We define two predicates, "x is a power of the prime v" and "x is a prime power" respectively:

$$\begin{aligned}\mathrm{pow}(x,v) &\equiv x \geq 1 \wedge \mathrm{prime}(v) \wedge \forall w \leq x(w > 1 \wedge w|x \to v|w)\\ \mathrm{pp}(x) &\equiv \exists v \leq x\, \mathrm{pow}(x,v)\end{aligned}$$

Exercise 58 a) $\mathrm{PA} \vdash \forall xv(\mathrm{pow}(x,v) \to \mathrm{pow}(x{\cdot}v,v))$

b) $\mathrm{PA} \vdash \forall xyv(\mathrm{pow}(x,v) \wedge \mathrm{pow}(y,v) \to x|y \vee y|x)$

c) $\mathrm{PA} \vdash \forall xyv(\mathrm{pow}(x,v) \wedge \mathrm{pow}(y,v) \wedge x < y \to (x{\cdot}v)|y)$

For prime(v), we want to define for each number $y > 0$ its v-part, that is the highest power of v that divides y. We denote this by $y \upharpoonright v$. For example, $12 \upharpoonright 3 = 3$, $12 \upharpoonright 2 = 4$ and $12 \upharpoonright 5 = 1$.

We assume as axiom:

$$\mathrm{pow}(y \upharpoonright v, v) \wedge (y \upharpoonright v)|y \wedge (y \upharpoonright v){\cdot}v \nmid y$$

Of course, to be able to do this we have to prove that

$$\mathrm{PA} \vdash \forall yv\exists!z((z = 0 \wedge (y = 0 \vee \neg\mathrm{prime}(v))) \vee \mathrm{pow}(z,v) \wedge z|y \wedge z{\cdot}v \nmid y)$$

If $\mathrm{pow}(y,v)$ take $z = y$. Otherwise, $\exists w \leq y(w|y \wedge v \nmid w)$ hence $\exists z \leq y \exists w \leq y(y = w{\cdot}z \wedge v \nmid w)$, so by LNP there is a least such z. Then $\mathrm{pow}(z,v)$ and $z|y$. If $z{\cdot}v|y$ so $y = w'{\cdot}z{\cdot}v = w{\cdot}z$, then $w'{\cdot}v = w$, contradiction with $v \nmid w$. So z exists; its uniqueness follows from the Exercise above.

The following lemma states that $x|y$ iff every prime power which divides x also divides y.

Lemma 4.6

$$\mathrm{PA} \vdash \forall xy(x|y \leftrightarrow \forall v \leq x(\mathrm{pp}(v) \wedge v|x \to v|y))$$

Proof. The direction from left to right is trivial, as is the case $y = 0 \vee x = 1$ in the other direction. For a contradiction, let $x > 1$ be least such that

$$\exists y \geq 1(\forall v \leq x(\mathrm{pp}(v) \wedge v|x \to v|y) \wedge x \nmid y)$$

and take the least such y. Its remainder on division by x satisfies the same property, so we may assume $y < x$. Let $x = a{\cdot}y + b$ with $0 \leq b < y$. If $0 < b$ we have a contradiction with the minimality of y. So $b = 0$ and $x = a{\cdot}y$.

Suppose $a > 1$. Then a has a prime divisor v by 4.5. Since $\text{pp}(v)$ and $v|x$, $v|y$. But now we have

$$\text{pp}((y \restriction v)\cdot v) \wedge (y \restriction v)\cdot v|x \wedge (y \restriction v)\cdot v \nmid y$$

which is a contradiction. ■

We can now define the least common multiple and greatest common divisor of two numbers, and prove their basic properties in PA.

Let $x, y \geq 1$. Since $x|x\cdot y$ and $y|x\cdot y$ there is a unique least $w > 0$ with $x|w \wedge y|w$; we denote this w by $\text{lcm}(x, y)$. Clearly, $\text{lcm}(x, y) \leq x\cdot y$.

Writing $x\cdot y = a\cdot\text{lcm}(x, y) + b$, $0 \leq b < \text{lcm}(x, y)$ we see that $x|b \wedge y|b$ so if $b > 0$ we get a contradiction with the minimality of $\text{lcm}(x, y)$. So $x\cdot y = a\cdot\text{lcm}(x, y)$ for a unique a, which we denote by $\text{gcd}(x, y)$. Writing $\text{lcm}(x, y) = y\cdot z$, we have $x\cdot y = \text{gcd}(x, y)\cdot y\cdot z$ so $x = \text{gcd}(x, y)\cdot z$ and $\text{gcd}(x, y)|x$; similarly, $\text{gcd}(x, y)|y$.

Exercise 59 Define yourself the function symbols $\max(x, y)$ and $\min(x, y)$ and prove their basic properties in PA. Prove furthermore:

a) $\text{PA} \vdash \text{prime}(v) \rightarrow \text{lcm}(x, y) \restriction v = \max(x \restriction v, y \restriction v)$

b) $\text{PA} \vdash \text{prime}(v) \rightarrow \text{gcd}(x, y) \restriction v = \min(x \restriction v, y \restriction v)$

Proposition 4.7

a) $\text{PA} \vdash \forall xyu(x, y \geq 1 \wedge x|u \wedge y|u \rightarrow \text{lcm}(x, y)|u)$

b) $\text{PA} \vdash \forall xyu(x, y \geq 1 \wedge u|x \wedge u|y \rightarrow u|\text{gcd}(x, y))$

Proof. For a), consider the remainder of u on division by $\text{lcm}(x, y)$; if it is non-zero, it is $< \text{lcm}(x, y)$ and still a common multiple of x and y.

For b), use proposition 4.6. Let $\text{pow}(z, v) \wedge z|u$. Then $z|(x \restriction v) \wedge z|(y \restriction v)$ so $z|(\text{gcd}(x, y) \restriction v)$ (by the Exercise), so $z|\text{gcd}(x, y)$. By 4.6, $u|\text{gcd}(x, y)$. ■

Exercise 60 Prove:

a) $\text{PA} \vdash \forall xy \geq 1 \forall x'y'(x = x'\cdot\text{gcd}(x, y) \wedge y = y'\cdot\text{gcd}(x, y) \rightarrow \text{gcd}(x', y') = 1)$

b) $\text{PA} \vdash \forall xyab(y = a\cdot x + b \wedge 0 \leq b < x \rightarrow \text{gcd}(x, y) = \text{gcd}(x, b))$

Theorem 4.8 (Bézout's Theorem for PA)

$$\text{PA} \vdash \forall xy \geq 1 \exists a \leq y, b \leq x (a{\cdot}x = b{\cdot}y + \gcd(x,y)))$$

Proof. By induction on x. For $x = 1$ take $a = 1, b = 0$.

For $x > 1$ let $y = c{\cdot}x + d$, $0 \leq d < x$. Dividing this equation by $\gcd(x, y)$ we have $y' = c{\cdot}x' + d'$ with $d' < x' \leq x$ and $\gcd(x', d') = 1$; by induction hypothesis we have

$$u{\cdot}d' = v{\cdot}x' + 1$$

for suitable u, v; so $v{\cdot}x' = u{\cdot}d' - 1$. Squaring both sides gives

$$a'{\cdot}x' = b'{\cdot}d' + 1$$

for some a', b'; multiplying by $\gcd(x, y)$ gives

$$(a' + b'{\cdot}c){\cdot}x = b'{\cdot}y + \gcd(x, y)$$

Finally, let $(a' + b'{\cdot}c) = c'{\cdot}y + a''$, $0 \leq a'' < y$. Then

$$a''{\cdot}x = (b' - c'{\cdot}x){\cdot}y + \gcd(x, y)$$

with $a'' < y$ and since $(b' - c'{\cdot}x){\cdot}y \leq a''{\cdot}x < x{\cdot}y$, we have $(b' - c'{\cdot}x) < x$. ∎

Theorem 4.8 plays a central role in the development of a rudimentary *coding of sequences* in PA, which was in fact Gödel's first crucial idea for the proof of his Incompleteness Theorems.

For a good understanding of what follows, it is useful first to see the algebraic trick underlying it. Suppose we are given a sequence of numbers $x_0, \ldots, x_{n-1}$.

Let $m = \max(x_0, \ldots, x_{n-1}, n)!$. Then for all i, j with $0 \leq i < j < n$ we have that the numbers $m(i+1)+1$ and $m(j+1)+1$ are relatively prime, for if p is a prime number which divides both of them, it divides their difference which is $m(j-i)$. Since p is prime, it follows that $p|m$, but also $p|(i+1)m+1$, a contradiction. Since $x_i < (i + 1)m + 1$ for all i, we have by the Chinese remainder theorem a number a such that

$$a \equiv x_i \bmod m(i + 1) + 1$$

for all i. The number a, or rather the pair (a, m), codes the sequence $x_0, \ldots, x_{n-1}$ in a sense.

The following theorem establishes three essential properties of this coding in PA: for every x, there is a sequence starting with x; every sequence can be extended; and a technical condition necessary later on.

We use the following abbreviations: $\text{rm}(x, y)$ denotes the remainder of x on division by y, and $(a, m)_i$ denotes $\text{rm}(a, m{\cdot}(i + 1) + 1)$.

Theorem 4.9

i) $\mathrm{PA} \vdash \forall x \exists a, m((a,m)_0 = x)$

ii) $\mathrm{PA} \vdash \forall yxam \exists bn(\forall i < y((a,m)_i = (b,n)_i) \wedge (b,n)_y = x)$

iii) $\mathrm{PA} \vdash \forall ami((a,m)_i \leq a)$

Proof. For i), take $m = x$ and $a = 2x+1$; then

$$\mathrm{rm}(a, m\cdot(0+1)+1) = \mathrm{rm}(2x+1, x+1) = x$$

iii) is trivial, so we are left to prove ii). Let us observe:

$$\begin{array}{ll} \mathrm{PA} \vdash \forall yxam \exists u(\forall i < y((a,m)_i < u) \wedge x < u \wedge y < u) & (1) \\ \mathrm{PA} \vdash \forall u \exists v \geq 1 \forall i \leq u\, (i \geq 1 \rightarrow i|v) & (2) \\ \mathrm{PA} \vdash \forall uv(\forall i \leq u(i \geq 1 \rightarrow i|v) \rightarrow & \\ \quad \forall ij(0 \leq i < j \leq u \rightarrow \gcd((i+1)\cdot v+1, (j+1)\cdot v+1) = 1)) & (3) \end{array}$$

((1) is proved by induction on y, (2) by induction on u, and (3) by formalizing the informal argument given above, using the properties about gcd that we know)

So, given y, x, a, m, take successively u satisfying (1) and v satisfying (2) for u; put $n = v$. We have:

$$\begin{gathered} \forall i < y((a,m)_i < (i+1)\cdot n+1) \\ x < (y+1)\cdot n+1 \\ \forall ij(0 \leq i < j \leq y \rightarrow \gcd((i+1)\cdot n+1, (j+1)\cdot n+1) = 1) \end{gathered}$$

and we want to find b such that

$$(\forall i < y((a,m)_i = (b,n)_i)) \wedge x = (b,n)_y$$

To do this we employ induction. Suppose for $k < y$ there is b' satisfying

$$(\forall i < k((a,m)_i = (b',n)_i)) \wedge x = (b',n)_y$$

We want to find b satisfying

$$(\forall i \leq k((a,m)_i = (b,n)_i)) \wedge x = (b,n)_y$$

Now it is easy to show that for all $k < y$,

$$\exists w((y+1)\cdot n+1|w \wedge \forall i < k((i+1)\cdot n+1|w) \wedge \gcd(w, (k+1)\cdot n+1) = 1)$$

(use induction on k and the properties of n). Take such w. Then by 4.8, there is $u \leq (k+1)\cdot n+1$ such that

$$\mathrm{rm}(u\cdot w, (k+1)\cdot n+1) = 1$$

Put $b = b' + u\cdot w\cdot(b'\cdot n\cdot(k+1) + (a,m)_k)$. Then $(b,n)_y = (b',n)_y = x$ since $(y+1)\cdot n+1|w$, and $i < k \rightarrow (b,n)_i = (b',n)_i = (a,m)_i$ since $(i+1)\cdot n+1|w$. Finally,

$$\begin{aligned}(b,n)_k &= \mathrm{rm}(b, (k+1)\cdot n+1)\\ &= \mathrm{rm}(b' + b'\cdot n\cdot(k+1) + (a,m)_k, (k+1)\cdot n+1)\\ &= \mathrm{rm}(b'\cdot((k+1)\cdot n+1) + (a,m)_k, (k+1)\cdot n+1)\\ &= (a,m)_k\end{aligned}$$

which completes the induction step and the proof. ∎

We shall shortly see (in Theorem 4.13 below) how to use theorem 4.9 to define every primitive recursive function in PA, after the necessary definitions to make precise what this means. But to give the idea already now, let's "define" the exponential function $x, y \mapsto x^y$. Let $\theta(x,y,z)$ be the formula

$$\exists am((a,m)_0 = 1 \wedge \forall i < y((a,m)_{i+1} = x\cdot(a,m)_i) \wedge (a,m)_y = z)$$

Exercise 61 Prove that $\mathrm{PA} \vdash \forall xy\exists!z\theta(x,y,z)$. Introduce a function symbol exp to $\mathcal{L}_{\mathrm{PA}}$, with axiom $\forall xy\theta(x,y,\exp(x,y))$. Prove:

$$\begin{aligned}&\mathrm{PA} \vdash \forall xyy'(\exp(x, y+y') = \exp(x,y)\cdot\exp(x,y'))\\ &\mathrm{PA} \vdash \forall xyy'(\exp(x, y\cdot y') = \exp(\exp(x,y), y'))\\ &\mathrm{PA} \vdash \forall xv(\mathrm{pow}(x,v) \rightarrow \exists y < x(x = \exp(v,y)))\end{aligned}$$

And try your hand at:

Exercise 62 Formulate and prove in PA the theorem of unique prime factorization.

4.2 Representing Recursive Functions in PA

Definition 4.10 An $\mathcal{L}_{\mathrm{PA}}$-formula φ is called a Δ_0-formula if all quantifiers are bounded in φ, that is of the form $\forall x < t$ or $\exists x < t$, for a term t not containing the variable x. A formula φ is a Σ_1-formula if it is of the form $\exists y_1 \ldots y_t\psi$ with ψ a Δ_0-formula. We also write $\varphi \in \Delta_0$, $\varphi \in \Sigma_1$.

Exercise 63 Prove the *Collection Principle* in PA:

$$\mathrm{PA} \vdash \forall i < t \exists v \psi \to \exists v \forall i < t \exists u < v \psi$$

and deduce that if φ is equivalent to a Σ_1-formula, so is $\forall i < t\varphi$.

We now discuss the so-called "Σ_1-completeness" of PA: the statement that PA proves all Σ_1-sentences which are true in the standard model $\mathcal{N}$. Recall the definition of the numerals $\overline{n}$ from page 56.

Exercise 64 Prove:

$$\begin{array}{rl}(\mathrm{PA} \vdash \overline{n} + \overline{m} = \overline{k}) \Leftrightarrow n + m = k & \text{for all } n, m, k \in \mathbb{N} \\ (\mathrm{PA} \vdash \overline{n} \cdot \overline{m} = \overline{k}) \Leftrightarrow n \cdot m = k & \text{for all } n, m, k \in \mathbb{N} \\ (\mathrm{PA} \vdash \overline{n} < \overline{m}) \Leftrightarrow n < m & \text{for all } n, m \in \mathbb{N} \\ \mathrm{PA} \vdash \forall x(x < \overline{n} \leftrightarrow x = \overline{0} \vee \ldots \vee x = \overline{n-1}) & \text{for all } n > 0\end{array}$$

From this exercise we can see by induction on the $\mathcal{L}_{\mathrm{PA}}$-term $t(x_1, \ldots, x_k)$ with variables $x_1, \ldots, x_k$: if $t^{\mathcal{N}}$ is its interpretation in the model $\mathcal{N}$, as function $\mathbb{N}^k \to \mathbb{N}$, then for all $n_1, \ldots, n_k \in \mathbb{N}$:

$$\mathrm{PA} \vdash t(\overline{n_1}, \ldots, \overline{n_k}) = \overline{t^{\mathcal{N}}(n_1, \ldots, n_k)}$$

Exercise 65 [Σ_1-completeness of PA] Prove that for every Δ_0-formula φ with free variables $x_1, \ldots, x_k$ and all $n_1, \ldots, n_k \in \mathbb{N}$:

$$\mathrm{PA} \vdash \varphi(\overline{n_1}, \ldots, \overline{n_k}) \Leftrightarrow \mathcal{N} \models \varphi[n_1, \ldots, n_k]$$

and deduce that the same equivalence holds for Σ_1-formulas. Conclude that a Σ_1-sentence is provable in PA if and only if it is true in $\mathcal{N}$.

Warning. The equivalence does *not* hold for negations of Σ_1-formulas, as we shall soon see!

Definition 4.11 Let $A \subseteq \mathbb{N}^k$ a k-ary relation. An $\mathcal{L}_{\mathrm{PA}}$-formula $\varphi(x_1, \ldots, x_k)$ of k free variables is said to *represent A (numeralwise)* if for all $n_1, \ldots, n_k \in \mathbb{N}$ we have:

$$\begin{array}{lll}(n_1, \ldots, n_k) \in A & \Rightarrow & \mathrm{PA} \vdash \varphi(\overline{n_1}, \ldots, \overline{n_k}) \quad \text{and} \\ (n_1, \ldots, n_k) \notin A & \Rightarrow & \mathrm{PA} \vdash \neg\varphi(\overline{n_1}, \ldots, \overline{n_k})\end{array}$$

Let $F : \mathbb{N}^k \to \mathbb{N}$ a k-ary function. An $\mathcal{L}_{\mathrm{PA}}$-formula $\varphi(x_1, \ldots, x_k, z)$ of $k+1$ free variables represents F numeralwise if for all $n_1, \ldots, n_k \in \mathbb{N}$:

$$\begin{array}{c}\mathrm{PA} \vdash \varphi(\overline{n_1}, \ldots, \overline{n_k}, \overline{F(n_1, \ldots, n_k)}) \quad \text{and} \\ \mathrm{PA} \vdash \exists! z \varphi(\overline{n_1}, \ldots, \overline{n_k}, z)\end{array}$$

Exercise 66 If $F : \mathbb{N}^k \to \mathbb{N}$ is numeralwise represented then so is its graph, considered as $k+1$-ary relation.

We say that a relation or function is Σ_1-represented if there is a Σ_1-formula representing it. Later, we shall see that if a function is represented at all, it must be Σ_1-represented, and recursive (and vice versa).

Definition 4.12 A function $F : \mathbb{N}^k \to \mathbb{N}$ is called *provably recursive* in PA if it is represented by a Σ_1-formula $\varphi(x_1, \ldots, x_k, z)$ for which

$$\mathrm{PA} \vdash \forall x_1 \ldots x_k \exists! z \varphi(x_1, \ldots, x_k, z)$$

Theorem 4.13 *Every primitive recursive function is provably recursive in* PA.

Proof. We prove this by induction on the generation of the primitive recursive function. The basic functions $\lambda x_1 \cdots x_k . x_i$, $\lambda x . x+1$ and $\lambda x . 0$ are clearly provably recursive.

If $F(\vec{x})$ is defined by composition from $G, H_1, \ldots, H_m$, so

$$F(\vec{x}) = G(H_1(\vec{x}), \ldots, H_m(\vec{x}))$$

suppose by induction hypothesis that $G, H_1, \ldots, H_m$ are represented by the Σ_1-formulas $\psi, \chi_1, \ldots, \chi_m$ respectively. Then F is represented by the formula

$$\varphi(\vec{x}, z) \equiv \exists z_1 \cdots z_m (\chi_1(\vec{x}, z_1) \wedge \cdots \wedge \chi_m(\vec{x}, z_m) \wedge \psi(z_1, \ldots, z_m, z))$$

which is equivalent to a Σ_1-formula; that $\mathrm{PA} \vdash \forall \vec{x} \exists! z \varphi(\vec{x}, z)$ follows from the corresponding property for $\psi, \chi_1, \ldots, \chi_m$.

The crucial induction step is primitive recursion; it is here that we use theorem 4.9. Suppose that $F(\vec{x}, y)$ is defined by primitive recursion from G and H, so

$$F(\vec{x}, 0) = G(\vec{x}) \text{ and } F(\vec{x}, y+1) = H(\vec{x}, F(\vec{x}, y), y)$$

By induction hypothesis, G and H are Σ_1-represented by $\psi(\vec{x}, z)$ and $\chi(\vec{x}, u, v, w)$ respectively. Then F is represented by the formula $\varphi(\vec{x}, y, u)$ defined as

$$\exists am(\psi(\vec{x}, (a,m)_0) \wedge \forall i < y \, \chi(\vec{x}, (a,m)_i, i, (a,m)_{i+1}) \wedge (a,m)_y = u)$$

To be sure, this should really be seen as an *abbreviation*, since there is no term $(a,m)_i$ in $\mathcal{L}_{\mathrm{PA}}$, so e.g. $\psi(\vec{x}, (a,m)_0)$ is shorthand for

$$\exists c, d < a(a = c{\cdot}(m+1) + d \wedge 0 \leq d < m+1 \wedge \psi(\vec{x}, d))$$

but still one sees that the formula φ is equivalent to a Σ_1-formula. The proof that PA $\vdash \forall\vec{x}, y\exists! u\varphi(\vec{x}, y, u)$ is done by induction (in PA!) on u, where one uses the properties listed in theorem 4.9. The details of this proof, as well as the proof that φ represents F, are left to the reader. ∎

Exercise 67 Carry out the filling in of missing details in the proof of theorem 4.13.

The study of the class of all functions which are provably recursive in PA, is important for the *proof theory* of PA. It is an old result that the provably recursive functions in PA are the ε_0*-recursive* functions. This refers to an ordinal hierarchy of total recursive functions, and ε_0 is the least ordinal α such that there exists a recursive binary relation $\prec$ on $\mathbb{N}$ with the properties:

- $(\mathbb{N}, \prec)$ is a well-order of order-type α;
- PA does not prove the scheme

$$\forall x(\forall y \prec x\psi(y) \to \psi(x)) \to \forall x\psi(x)$$

(where, of course, we use a Σ_1-formula representing $\prec$ in PA)

There are several equivalent definitions of ε_0; another one is: the least ordinal which is closed under the operation $\beta \mapsto \omega^\beta$.

We do not enter this study in this course, but just point out that there are lots of provably total functions which are not primitive recursive. To give the simplest possible case:

Exercise 68 Prove that the Ackermann function:

$$\begin{array}{rcl} A(0, x) & = & x+1 \\ A(n+1, 0) & = & A(n, 1) \\ A(n+1, x+1) & = & A(n, A(n+1, x)) \end{array}$$

is provably recursive in PA.

Theorem 4.14 *Every total recursive function is* Σ_1*-represented in* PA*.*

Proof. By basic recursion theory, there is a primitive recursive predicate T, a primitive recursive function U such that for every k-ary recursive function F we have a number e such that:

$$F(n_1, \ldots, n_k) = m \Leftrightarrow \exists y(T(e, n_1, \ldots, n_k, y) \wedge U(y) = m)$$

The set $\{(n_1,\ldots,n_k,y,m)\,|\,T(e,n_1,\ldots,n_k,y)\wedge U(y)=m\}$ is primitive recursive and so, by 4.13, represented by a Σ_1-formula $\varphi(x_1,\ldots,x_k,y,w)$, which we can write as

$$\exists z_1\ldots z_l P(x_1,\ldots,x_k,y,w,z_1,\ldots,z_l)$$

for a Δ_0-formula P.

If $R(z,\vec{x},w)$ is the Δ_0-formula $\exists y<z\exists z_1<z\cdots\exists z_l<zP$, then clearly

$$\mathrm{PA}\vdash\exists yw\varphi(\vec{x},y,w)\leftrightarrow\exists zwR(z,\vec{x},w)$$

Finally, let $S(z,\vec{x},w)$ be the Δ_0-formula

$$w<z\wedge R(z,\vec{x},w)\wedge\forall u<z\neg\exists v<uR(u,\vec{x},v)$$

Then $\mathrm{PA}\vdash\exists zwR(z,\vec{x},w)\leftrightarrow\exists!z\exists wS(z,\vec{x},w)$ by LNP.

I claim that the Σ_1-formula $\exists zS(z,\vec{x},w)$ represents the function F. First, for $n_1,\ldots,n_k\in\mathbb{N}$ is

$$\exists zS(z,\overline{n_1},\ldots,\overline{n_k},\overline{F(n_1,\ldots,n_k)})$$

a true Σ_1-formula, hence provable in PA by Σ_1-completeness. To show that

$$\mathrm{PA}\vdash\exists!w\exists zS(z,\overline{n_1},\ldots,\overline{n_k},w)$$

let $a\in\mathbb{N}$ such that $S(\overline{a},\overline{n_1},\ldots,\overline{n_k},\overline{F(n_1,\ldots,n_k)})$ is true. By unicity of z in S we have

$$\mathrm{PA}\vdash\forall zw(S(z,\overline{n_1},\ldots,\overline{n_k},w)\to z=\overline{a}\wedge w<\overline{a})$$

and since $\mathrm{PA}\vdash\forall w<\overline{a}\,(w=\overline{0}\vee\cdots\vee w=\overline{a-1})$, we have

$$\begin{array}{rl}\mathrm{PA}\vdash\overline{F(n_1,\ldots,n_k)}<\overline{a} & \text{and}\\ \mathrm{PA}\vdash\neg S(\overline{a},\overline{n_1},\ldots,\overline{n_k},\overline{b}) & \text{for all } b<a,\ b\neq F(n_1,\ldots,n_k)\end{array}$$

since $S\in\Delta_0$. So, $\mathrm{PA}\vdash\exists!w\exists zS(z,\overline{n_1},\ldots,\overline{n_k},w)$. ■

Exercise 69 In the next chapter we shall see that there are Σ_1-sentences which are false in $\mathcal{N}$ but consistent with PA. Use this to show that the following implication does *not* hold: for a Σ_1-formula $\varphi(w)$ with only free variable w, if $\exists!w\varphi(w)$ is true in $\mathcal{N}$, then $\mathrm{PA}\vdash\exists!w\varphi(w)$.

Exercise 70 Prove that every recursive set is Σ_1-represented in PA.

Exercise 71 Let $D_1, D_2, D_3, \ldots$ be a sequence of definitions of primitive recursive functions with the properties that for every k, the function f_k defined by D_k is either a basic function or defined from functions f_l with $l < k$, and every primitive recursive function is f_k for some k.

Introduce, for every k, a new function symbol F_k and an axiom φ_k, corresponding to the definition D_k of f_k.

Let PA$'$ be the theory in the language $\mathcal{L}_{\mathrm{PA}} \cup \{F_1, F_2, \ldots\}$, axiomatized by the axioms of PA, together with the axioms φ_k, and the scheme of induction extended to the full new language.

Prove that there is a mapping $(\cdot)^*$ from $\mathcal{L}_{\mathrm{PA}'}$-formulas to $\mathcal{L}_{\mathrm{PA}}$-formulas, which is the identity on $\mathcal{L}_{\mathrm{PA}}$-formulas, such that

$$\begin{array}{c} \mathrm{PA}' \vdash \varphi \leftrightarrow (\varphi)^* \\ \mathrm{PA}' \vdash \varphi \Rightarrow \mathrm{PA} \vdash (\varphi)^* \end{array}$$

for all $\mathcal{L}_{\mathrm{PA}'}$-formulas φ. Conclude that PA$'$ is *conservative* over PA: this means that every L_{PA}-sentence which is provable in PA$'$, is provable in PA.

Exercise 72 Devise a coding of the definitions D_k in the previous exercise, and show that a *recursive* sequence $D_1, D_2, \ldots$ exists with the required properties. Can it be primitive recursive?

4.2.1 The 'Entscheidungsproblem'

The *Entscheidungsproblem* (decision problem) was posed by Hilbert and Ackermann in [13]. In modern terms, the question is: is there an algorithm which decides whether a given formula in predicate logic (as we have formulated it in chapter 1) is valid?

It was Alonzo Church ([3]) who noted that as a consequence of the theory developed in this chapter, a negative answer can be given to this question (provided we take theorem 3.24, which we have formulated without proof, for granted).

Let F denote the primitive recursive function defined by:

$$F(e,x,y) \;=\; \left\{ \begin{array}{ll} 0 & \text{if } T(1,e,x,y) \\ 1 & \text{else} \end{array} \right.$$

where T is the Kleene T-predicate.
Then F is provably recursive in PA by theorem 4.13; let $\chi(e,x,y,n)$ be a Σ_1-formula representing F. Then in the proof that χ represents F and

represents a total function in PA, we have employed a finite number of induction axioms. Also, the proof of Σ_1-completeness for PA uses finitely many induction axioms. Let S be the subtheory of PA consisting of all those induction axioms, together with the first 6 axioms of PA. Then S is a finite theory; and for any sentence ϕ of $\mathcal{L}_{\mathrm{PA}}$, we have that ϕ is a consequence of S if and only if the sentence $(\bigwedge_{\psi \in S} \psi) \to \phi$ is valid in the predicate calculus.

Therefore, if we can show that there can be no algorithm which decides for such ϕ whether or not ϕ is a consequence of S, we have proved that the Entscheidungsproblem is unsolvable.

We have, for arbitrary numbers e and x, the following equivalences:

$$\begin{array}{rcl} x \in \mathrm{dom}(\varphi_e^{(1)}) & \Leftrightarrow & \text{by Chapter 3} \\ \text{there is a } y \text{ such that } F(e,x,y)=0 & \Leftrightarrow & \text{since } \chi \text{ represents } F \\ \mathcal{N} \models \exists y \chi(\bar{e},\bar{x},y,0) & \Leftrightarrow & \text{by } \Sigma_1\text{-completeness} \\ S \models \exists y \chi(\bar{e},\bar{x},y,0) & & \end{array}$$

Therefore, any algorithm which decides whether or not a given $\mathcal{L}_{\mathrm{PA}}$-sentence is a consequence of S, gives us an algorithm which decides whether or not $x \in \mathrm{dom}(\varphi_e^{(1)})$. But this means that this latter set is recursive, which contradicts theorem 3.24.

4.3 A Primitive Incompleteness Theorem

The representability of recursive functions allows us to prove already that PA is not a complete theory: there is an L_{PA}-sentence ϕ such that PA $\not\vdash \phi$ and PA $\not\vdash \neg\phi$ (this, however, is not quite Gödel's theorem; the latter gives more information). We have to leave one detail to the reader's imagination (it will be fully treated in the next chapter, but it is easy): for every $\mathcal{L}_{\mathrm{PA}}$-formula $\varphi(w)$ with exactly one free variable w, the set

$$\{n \in \mathbb{N} \,|\, \mathrm{PA} \vdash \varphi(\overline{n})\}$$

is *recursively enumerable.*

Now we do know, that for every recursively enumerable set $X \subseteq \mathbb{N}$, there is a Σ_1-formula $\varphi(w)$, such that for all $n \in \mathbb{N}$:

$$n \in X \Leftrightarrow \mathrm{PA} \vdash \varphi(\overline{n})$$

(Use the characterization of r.e. sets as projections of recursive sets, representability of recursive sets in PA, and Σ_1-completeness of PA)

Now, let X be a nonrecursive, r.e. set (which exists by Theorem 3.24) and suppose the Σ_1-sentence φ defines X in this sense. Let $Y = \{n \in \mathbb{N} \mid \mathrm{PA} \vdash \neg\varphi(\overline{n})\}$. Then since PA is consistent, X and Y are disjoint r.e. sets and since X is not recursive, Y is not the complement of X (by Exercise 51). Take $m \notin X \cup Y$. Since $\mathrm{PA} \vdash \varphi(\overline{m})$ implies $m \in X$ and $\mathrm{PA} \vdash \neg\varphi(\overline{m})$ implies $m \in Y$, we see that none of these can hold; therefore, $\varphi(\overline{m})$ is a sentence which is independent of PA.

The following exercise is a result which will be needed in the next chapter. We call a formula $\varphi(x_1, \ldots, x_k)$ Δ_1, or *a* Δ_1*-formula*, if both φ and $\neg\varphi$ are equivalent (in PA) to a Σ_1-formula.

Exercise 73 Show that the proof of theorem 4.13 can be adapted to give the following stronger result: for every primitive recursive function $F : \mathbb{N}^k \to \mathbb{N}$ there is a Δ_1-formula $\varphi_F(x_1, \ldots, x_{k+1})$ which represents F and is such that

$$\mathrm{PA} \vdash \forall x_1 \cdots x_k \exists! x_{k+1} \varphi_F(x_1, \ldots, x_{k+1})$$

Chapter 5

Gödel Incompleteness

5.1 Coding of Formulas and Diagonalization

We start by applying the primitive recursive coding of sequences from Chapter 3 to code formulas of PA.

We use sequence encoding to assign to any formula φ of $\mathcal{L}_{\mathrm{PA}}$ a code $\ulcorner\varphi\urcorner \in \mathbb{N}$ and this in such a way that all relevant operations on formulas translate into primitive recursive functions on codes.

We assume that in our language, variables are numbered $v_0, v_1, \ldots$. Consider the following "code book" (from now on we take $<$ as a primitive symbol of $\mathcal{L}_{\mathrm{PA}}$):

0	1	v	$+$	$\cdot$	$=$	$<$	$\wedge$	$\vee$	$\rightarrow$	$\neg$	$\forall$	$\exists$
0	1	2	3	4	5	6	7	8	9	10	11	12

For each term t define its code $\ulcorner t\urcorner$ by recursion on t: $\ulcorner 0\urcorner = \langle 0\rangle$, $\ulcorner 1\urcorner = \langle 1\rangle$, $\ulcorner v_i\urcorner = \langle 2, i\rangle$; $\ulcorner t+s\urcorner = \langle 3, \ulcorner t\urcorner, \ulcorner s\urcorner\rangle$, $\ulcorner t\cdot s\urcorner = \langle 4, \ulcorner t\urcorner, \ulcorner s\urcorner\rangle$.

It is now immediate that the properties "x is the code of a term", "x codes a constant", "the variable v_i occurs in the term coded by x", etcetera, are all primitive recursive in their arguments.

Likewise, we define codes for formulas: $\ulcorner t = s\urcorner = \langle 5, \ulcorner t\urcorner, \ulcorner s\urcorner\rangle$, $\ulcorner t < s\urcorner = \langle 6, \ulcorner t\urcorner, \ulcorner s\urcorner\rangle$, $\ulcorner \varphi \wedge \psi\urcorner = \langle 7, \ulcorner \varphi\urcorner, \ulcorner \psi\urcorner\rangle$, $\ulcorner \varphi \vee \psi\urcorner = \langle 8, \ulcorner \varphi\urcorner, \ulcorner \psi\urcorner\rangle$ and so on; $\ulcorner \forall v_i \varphi\urcorner = \langle 11, i, \ulcorner \varphi\urcorner\rangle$ and $\ulcorner \exists v_i \varphi\urcorner = \langle 12, i, \ulcorner \varphi\urcorner\rangle$.

And we have that the properties "x codes a formula", "the main connective of the formula coded by x is $\wedge$", "the variable v_i occurs freely in the formula coded by x" and so forth, are primitive recursive in their arguments.

Exercise 74 Verify this for some of the mentioned properties.

Exercise 75 Verify that the property "x codes a formula φ and y codes a term t and t is free for v_i in φ" is primitive recursive in x, y, i; and show that there is a primitive recursive function Sub, such that

$$\mathrm{Sub}(x,y,i) = \begin{cases} \ulcorner \varphi[s/v_i] \urcorner & \text{if } y = \ulcorner \varphi \urcorner \text{ and } x = \ulcorner s \urcorner \\ 0 & \text{else} \end{cases}$$

Exercise 76 Convince yourself that the properties "x is the code of a Δ_0-formula" and "x codes a Σ_1-formula" are primitive recursive.

Having done this work, we now arrive at the *second* main idea of Gödel, the Diagonalization Lemma.
We say that φ is a Π_1-formula if it is of the form $\forall y_1 \cdots \forall y_n \psi$ with $\psi \in \Delta_0$.

Lemma 5.1 (Diagonalization Lemma) *For any $\mathcal{L}_{\mathrm{PA}}$-formula φ with free variable v_0 there is an $\mathcal{L}_{\mathrm{PA}}$-formula ψ with the same free variables as φ except v_0, such that*

$$\mathrm{PA} \vdash \psi \leftrightarrow \varphi[\overline{\ulcorner \psi \urcorner}/v_0]$$

Moreover, if $\varphi \in \Pi_1$ then ψ can be chosen to be Π_1 too.

Proof. Recall the function $\mathrm{Sub}(x,y,i)$ from Exercise 75. It is primitive recursive hence so is $\lambda xy.\mathrm{Sub}(x,y,0)$; let S be a Σ_1-formula representing this function in PA. Let T be a Σ_1-formula representing the primitive recursive function $n \mapsto \ulcorner \overline{n} \urcorner$. Then we have

$$\begin{array}{cr} \forall nm \in \mathbb{N}.\mathrm{PA} \vdash S(\overline{n}, \overline{m}, \overline{\mathrm{Sub}(n,m,0)}) & (1) \\ \forall n \in \mathbb{N}.\mathrm{PA} \vdash T(\overline{n}, \overline{\ulcorner \overline{n} \urcorner}) & (2) \\ \mathrm{PA} \vdash \forall xy \exists ! z S(x,y,z) & (3) \\ \mathrm{PA} \vdash \forall x \exists ! y T(x,y) & (4) \end{array}$$

Now let φ have v_0 free. Define the formula C by

$$C \equiv \forall xy (T(v_0, x) \wedge S(x, v_0, y) \to \varphi[y/v_0])$$

and let ψ be defined by

$$\psi \equiv C[\overline{\ulcorner C \urcorner}/v_0] \qquad (5)$$

Clearly, if $\varphi \in \Pi_1$ then so are C and ψ. Now we have by (2) and (4),

$$\mathrm{PA} \vdash \forall y(\exists x(T(\overline{\ulcorner C \urcorner}, x) \wedge S(x, \overline{\ulcorner C \urcorner}, y)) \leftrightarrow S(\overline{\ulcorner \overline{\ulcorner C \urcorner} \urcorner}, \overline{\ulcorner C \urcorner}, y))$$

and (1) and (3) give us

$$\mathrm{PA} \vdash \forall y(S(\overline{\ulcorner \overline{\ulcorner C \urcorner} \urcorner}, \overline{\ulcorner C \urcorner}, y) \leftrightarrow y = \overline{\ulcorner C[\overline{\ulcorner C \urcorner}/v_0] \urcorner})$$

By (5) then,

$$\mathrm{PA} \vdash \forall y(\exists x(T(\overline{\ulcorner C \urcorner}, x) \wedge S(x, \overline{\ulcorner C \urcorner}, y)) \leftrightarrow y = \overline{\ulcorner \psi \urcorner})$$

so

$$\begin{array}{rcl} \mathrm{PA} \vdash \psi & \leftrightarrow & \forall y(\exists x(T(\overline{\ulcorner C \urcorner}, x) \wedge S(x, \overline{\ulcorner C \urcorner}, y)) \to \varphi[y/v_0]) \\ & \leftrightarrow & \forall y(y = \overline{\ulcorner \psi \urcorner} \to \varphi[y/v_0]) \\ & \leftrightarrow & \varphi[\overline{\ulcorner \psi \urcorner}/v_0] \end{array}$$

■

Remark. One should compare the proof of Lemma 5.1 with the proofs of very similar theorems, such as the recursion theorem (or, if you are familiar with it, the fixpoint theorem in λ-calculus).

I include the following corollary, which is analogous to Smullyan's simultaneous recursion theorem (Exercise 49), or Bekić' Lemma in Domain Theory, for its own interest. We shall not apply it.

Corollary 5.2 (Simultaneous Diagonalization) *Let φ and ψ be formulas both having the variables v_0, v_1 free. Then there are formulas θ and χ, such that θ has the same free variables as φ minus v_0, v_1, and ditto for χ and ψ, such that*

$$\mathrm{PA} \vdash \theta \leftrightarrow \varphi[\overline{\ulcorner \theta \urcorner}/v_0, \overline{\ulcorner \chi \urcorner}/v_1]$$
$$\mathrm{PA} \vdash \chi \leftrightarrow \psi[\overline{\ulcorner \theta \urcorner}/v_0, \overline{\ulcorner \chi \urcorner}/v_1]$$

And, if $\varphi, \psi \in \Pi_1$, so are θ, χ.

Proof. Let T be the same formula as in the proof of Lemma 5.1, and S_1 similar, that is: S_1 now represents substitution for the variable v_1. So $\mathrm{PA} \vdash S_1(\overline{\ulcorner s \urcorner}, \overline{\ulcorner \varphi \urcorner}, \overline{\ulcorner \varphi[s/v_1] \urcorner})$, etc. Let φ and ψ be given. First, apply Lemma 5.1 to find θ_1 such that

$$\mathrm{PA} \vdash \theta_1 \leftrightarrow \forall zy(T(v_1, z) \wedge S_1(z, \overline{\ulcorner \theta_1 \urcorner}, y) \to \varphi[y/v_0, v_1])$$

and then χ such that

$$\mathrm{PA} \vdash \chi \leftrightarrow \forall xy(T(\overline{\ulcorner \chi \urcorner}, z) \wedge S_1(z, \overline{\ulcorner \theta_1 \urcorner}, y) \to \psi[y/v_0, \overline{\ulcorner \chi \urcorner}/v_1])$$

Put $\theta \equiv \theta_1[\overline{\ulcorner\chi\urcorner}/v_1]$. Then as in the proof of Lemma 5.1, we have:

$$\begin{gathered}
\mathrm{PA} \vdash T(\overline{\ulcorner\chi\urcorner}, \overline{\ulcorner\overline{\ulcorner\chi\urcorner}\urcorner}) \wedge S_1(\overline{\ulcorner\overline{\ulcorner\chi\urcorner}\urcorner}, \overline{\ulcorner\theta_1\urcorner}, \overline{\ulcorner\theta_1[\overline{\ulcorner\chi\urcorner}/v_1]\urcorner}) \\
\mathrm{PA} \vdash \forall y(\exists z(T(\overline{\ulcorner\chi\urcorner}, z) \wedge S_1(z, \overline{\ulcorner\theta_1\urcorner}, y)) \leftrightarrow y = \overline{\ulcorner\theta\urcorner}) \\
\mathrm{PA} \vdash \theta \leftrightarrow \theta_1[\overline{\ulcorner\chi\urcorner}/v_1] \leftrightarrow \varphi[\overline{\ulcorner\theta\urcorner}, \overline{\ulcorner\chi\urcorner}]
\end{gathered}$$

and so, also $\mathrm{PA} \vdash \chi \leftrightarrow \psi[\overline{\ulcorner\theta\urcorner}, \overline{\ulcorner\chi\urcorner}]$. ∎

5.2 Gödel's First Incompleteness Theorem

Just as we have coded formulas, we can code proofs in PA by natural numbers. Since the idea is essentially the same, we give only a sketch. We use natural deduction. Again we make a code book, now of construction steps for natural deduction trees (I have *not* tried to make the system as economical as possible!):

Ass	0	$\vee$I − r	5	$\forall$E	10	$\bot$	15
$\wedge$I	1	$\vee$I − l	6	$\exists$I	11	$\neg\neg$	16
$\wedge$E − r	2	$\rightarrow$ E	7	$\exists$E	12		
$\wedge$E − l	3	$\rightarrow$ I	8	$\neg$I	13		
$\vee$E	4	$\forall$I	9	$\neg$E	14		

We view natural deduction proofs as labelled trees; every node is labelled by a formula, and by a rule. Most connectives have an *introduction* and an *elimination* rule, sometimes more than one, for example the rule $\wedge$E − r (conjunction elimination to the right) infers ψ from $\phi \wedge \psi$. The rule $\neg$E infers $\bot$ from $\phi, \neg\phi$; the rule $\bot$ infers ψ from $\bot$, the rule $\neg\neg$ infers ψ from $\neg\neg\psi$. The rule Ass (assumption) is the only starting rule: it allows one to construct a one-node tree, labelled with a formula φ. I hope that the meaning of every rule is now clear.

Now every tree has a set of so-called *open* (or undischarged) assumptions. An assumption is a formula which labels a leaf of the tree. Assumptions are discharged with the steps $\rightarrow$ I, $\neg$I, $\vee$E and $\exists$E. We follow the so-called *crude discharge convention*: that is, whenever we introduce $\varphi \rightarrow \psi$ by $\rightarrow$ I, we discharge *all* assumptions φ above this application.

Let us outline the coding of trees. The tree with one node, labelled φ, gets code $\langle 0, \ulcorner\varphi\urcorner\rangle$; suppose D_1, D_2 are trees with roots labelled by φ, ψ respectively; the tree resulting from D_1 and D_2 by applying $\wedge$I gets code $\langle 1, \ulcorner D_1\urcorner, \ulcorner D_2\urcorner, \ulcorner\varphi \wedge \psi\urcorner\rangle$, where $\ulcorner D_1\urcorner$ denotes the code of D_1. If D_2 results from D_1 by applying $\wedge E - r$, so the root of D_1 is labelled $\varphi \wedge \psi$ and the root of D_2 is labelled ψ, we have $\ulcorner D_2 \urcorner = \langle 2, \ulcorner D_1 \urcorner, \ulcorner \psi \urcorner \rangle$. If D_4 results from D_1, D_2, D_3 by $\vee$-elimination, that is: the root of D_1 is labelled $\varphi \vee \psi$, D_2 and D_3 have χ at the root, and D_4 also has χ at the root, whereby in D_2, all open assumptions φ are discharged and in D_3 all open assumptions ψ are discharged, we have $\ulcorner D_4 \urcorner = \langle 4, \ulcorner D_1 \urcorner, \ulcorner D_2 \urcorner, \ulcorner D_3 \urcorner, \ulcorner \chi \urcorner \rangle$.

I hope the process is now clear: the length of $\ulcorner D \urcorner$ is $n+2$ where n is the number of branches from the root (in fact, always $n \leq 3$), the first element of $\ulcorner D \urcorner$ is the code of the last rule applied, and the last element of $\ulcorner D \urcorner$ is the formula which labels the root of D. In this way, we can easily recover the whole tree D from its code $\ulcorner D \urcorner$. We can also define a primitive recursive function OA, which, given $\ulcorner D \urcorner$, gives a code for the set of undischarged assumptions of D. Therefore, we can, primitive recursively, check whether D is in fact a correct proof tree (for example, when introducing $\forall u \varphi(u)$ by $\forall$I from $\varphi(v)$, we need to know that the variable v does not occur in any undischarged assumption, and so on). The conclusion is that we have a primitive recursive predicate $\mathrm{NDT}(x,y)$: $\mathrm{NDT}(x,y)$ says that y is the code of a formula and x is the code of a correct natural deduction tree with root labelled by the formula coded by y.

In order that x codes a proof in PA, we need to know that all open assumptions of the tree coded by x are axioms of PA, or axioms of the predicate calculus governing the equality sign $=$: the axioms $u = u$, $u = v \wedge v = w \to u = w$ and $t = s \wedge \varphi[t/u] \to \varphi[s/u]$ (subject to the well-known conditions).

Exercise 77 Show that the predicate $\mathrm{Ax}(x)$: x is the code of an axiom of PA or the predicate calculus, is primitive recursive.

Let $\mathrm{Prf}(x,y)$ be the predicate: y is the code of a formula, and x is the code of a correct proof in PA of the formula coded by y:

$$\mathrm{Prf}(x,y) \leftrightarrow \mathrm{NDT}(x,y) \wedge \forall z \in \mathrm{OA}(x) \mathrm{Ax}(z)$$

Let $\overline{\mathrm{Prf}}$, $\overline{\mathrm{NDT}}$ and $\overline{\mathrm{Ax}}$ be Δ_1-formulas representing the predicates Prf, NDT, Ax in PA.

The predicate Prf is defined by a course-of-values recursion, and we can assume that PA proves this course of values recursion for the representing formula $\overline{\mathrm{Prf}}$. That is,

$$\mathrm{PA} \vdash \overline{\mathrm{Prf}}(x,y) \leftrightarrow C_0(x,y) \vee \cdots \vee C_{16}(x,y)$$

(referring to our code book of natural deduction rules), where $C_0(x, y)$ is the formula

$$x = \langle 0, y\rangle \wedge \overline{\mathrm{Ax}}(y)$$

$C_1(x, y)$ will be the formula

$$\exists abvw < x(y = \langle \bar{7}, v, w\rangle \wedge \overline{\mathrm{Prf}}(a, v) \wedge \overline{\mathrm{Prf}}(b, w) \wedge x = \langle \bar{1}, a, b, y\rangle)$$

and so on. In some cases, where open assumptions are discharged, we have to write conditions; e.g., C_8 (corresponding to → I) will read:

$$\exists avw < x(x = \langle \bar{8}, a, y\rangle \wedge y = \langle \bar{9}, v, w\rangle \wedge \overline{\mathrm{NDT}}(a, w) \wedge \forall z \in \mathrm{OA}(a)(\overline{\mathrm{Ax}}(z) \vee z = v))$$

(slightly abusing notation: "$z \in \mathrm{OA}(a)$" means of course the intended formalization)
It is now straightforward to see that we have the following proposition:

Proposition 5.3

i) $\mathrm{PA} \vdash \varphi \Rightarrow \mathrm{PA} \vdash \exists x \overline{\mathrm{Prf}}(x, \overline{\ulcorner \varphi \urcorner})$

ii) $\mathrm{PA} \vdash \forall xy(\overline{\mathrm{Prf}}(x, \overline{\ulcorner \varphi \to \psi \urcorner}) \wedge \overline{\mathrm{Prf}}(y, \overline{\ulcorner \psi \urcorner}) \to \overline{\mathrm{Prf}}(\langle \bar{7}, x, y, \overline{\ulcorner \psi \urcorner}\rangle, \overline{\ulcorner \psi \urcorner}))$

We introduce an abbreviation: $\Box\varphi$ for $\exists x \overline{\mathrm{Prf}}(x, \overline{\ulcorner \varphi \urcorner})$. Proposition 5.3 now says:

D1	$\mathrm{PA} \vdash \varphi \Rightarrow \mathrm{PA} \vdash \Box\varphi$
D2	$\mathrm{PA} \vdash \Box\varphi \wedge \Box(\varphi \to \psi) \to \Box\psi$

Theorem 5.4 (Gödel's First Incompleteness Theorem) *Apply Lemma 5.1 to the formula* $\neg\exists x \overline{\mathrm{Prf}}(x, v_0)$*, to obtain a* Π_1*-sentence* G *such that*

$$\mathrm{PA} \vdash G \leftrightarrow \neg\Box G$$

Then G is independent of PA.

Proof. Since $\overline{\mathrm{Prf}}(x, y)$ is Δ_1, clearly G can be chosen to be Π_1. If $\mathrm{PA} \vdash G$ then by D1, $\mathrm{PA} \vdash \Box G$, so $\mathrm{PA} \vdash \neg G$ by the choice of G. So PA is inconsistent, quod non.

On the other hand, if $\mathrm{PA} \vdash \neg G$ then $\mathrm{PA} \vdash \Box G$ by the choice of G. Then $\Box G$ is true in $\mathcal{N}$, which means that there is a proof of G, i.e. $\mathrm{PA} \vdash G$, and again PA is inconsistent. ∎

Remarks.

i) The sentence G is the famous "Gödel sentence". Roughly speaking it says "I am not provable", and it has therefore been compared with several liar paradoxes (see the work by Smullyan and Smorynski).

ii) The sentence G is true in $\mathcal{N}$, because if it were false, then $\neg G$ would be a true Σ_1-sentence, hence provable in PA by Σ_1-completeness.

iii) In the proof of Theorem 5.4, we have used the reasoning: "if PA $\vdash \varphi$ then $\mathcal{N} \models \varphi$" (in fact, we only used this for the Σ_1-sentence $\neg G$). This is not satisfactory, because we would like to extend Gödel's method to consistent extensions of PA, which need not have this property, even for Σ_1-sentences (for example, PA $\cup \{\neg G\}$ is such a theory). A way of avoiding this reasoning was found by Rosser, a few years after Gödel. Let $\varphi(v_0)$ be the formula

$$\forall x(\overline{\mathrm{Prf}}(x, v_0) \rightarrow \exists y < x\overline{\mathrm{Prf}}(y, \langle 10, v_0 \rangle))$$

Check that $\varphi(v_0)$ is equivalent to a Π_1-formula! Apply Lemma 5.1 to $\varphi(v_0)$, to obtain a Π_1-sentence R such that

$$\mathrm{PA} \vdash R \leftrightarrow \forall x(\overline{\mathrm{Prf}}(x, \overline{\ulcorner R \urcorner}) \rightarrow \exists y < x\overline{\mathrm{Prf}}(y, \overline{\ulcorner \neg R \urcorner}))$$

We can show that R is independent of PA, just using that PA is consistent and Σ_1-complete. Suppose PA $\vdash R$. By consistency of PA, PA $\nvdash \neg R$, whence the sentence

$$\exists x(\overline{\mathrm{Prf}}(x, \overline{\ulcorner R \urcorner}) \wedge \forall y < x\neg\overline{\mathrm{Prf}}(y, \overline{\ulcorner \neg R \urcorner}))$$

is a true Σ_1-sentence, hence by Σ_1-completeness provable in PA. But this sentence is equivalent to $\neg R$, contradiction. Conversely, if PA $\vdash$ $\neg R$ we have for some $n \in \mathbb{N}$ that PA $\vdash \overline{\mathrm{Prf}}(\bar{n}, \overline{\ulcorner \neg R \urcorner})$ and PA $\vdash \forall y < \bar{n}\neg\overline{\mathrm{Prf}}(y, \overline{\ulcorner R \urcorner})$, since these are true Σ_1-sentences. It follows that PA $\vdash$ $\forall x(\overline{\mathrm{Prf}}(x, \overline{\ulcorner R \urcorner}) \rightarrow \exists y < x\overline{\mathrm{Prf}}(y, \overline{\ulcorner \neg R \urcorner}))$, that is PA $\vdash R$. Again, a contradiction with the consistency of PA.

iv) As a consequence of the previous item, we can apply Gödel's method to finite (consistent) extensions of PA. This can be used to give a formulation of Gödel's theorem which does not even need the consistency of PA.

Call a partial order *dense* if whenever $x < y$, there is a z with $x < z < y$.

The *Lindenbaum algebra* of PA is the set of $\mathcal{L}_{\mathrm{PA}}$-sentences modulo PA-provable equivalence. Denote the equivalence class of ϕ by $[\phi]$. The Lindenbaum algebra is ordered by: $[\phi] \leq [\psi]$ iff $\mathrm{PA} \vdash \phi \to \psi$. With this ordering, the Lindenbaum algebra of PA is a Boolean algebra, with least element $\bot$ and top element $\neg\bot$.

We claim that *the Lindenbaum algebra of* PA *is dense.* Note that this certainly holds if PA is inconsistent, because then the algebra has only one element, and every one-element poset is trivially dense.

Suppose $[\phi] < [\psi]$, so $\mathrm{PA} \vdash \phi \to \psi$ and $\mathrm{PA} \not\vdash \psi \to \phi$. Then the theory $T = \mathrm{PA} \cup \{\psi, \neg\phi\}$ is consistent, so we can apply the Gödel method to it, and find a sentence ρ which is independent of T. Now let χ be the sentence $(\psi \wedge \rho) \vee \phi$. We leave it to you to check that indeed $[\phi] < [\chi] < [\psi]$. So, the Lindenbaum algebra is dense.

Now conversely, if the Lindenbaum algebra is dense, we can apply the denseness property (in the case that PA is consistent) to the inequality $[\bot] < [\neg\bot]$ to find a sentence ϕ such that $[\bot] < [\phi] < [\neg\bot]$; but then ϕ is independent of PA. So the denseness of the Lindenbaum algebra implies (if PA is consistent) Gödel's theorem.

v) The sentence $\neg\Box\bot$ is called the sentence expressing the consistency of PA, and often written as $\mathrm{Con}_{\mathrm{PA}}$. It is an easy consequence of D2 that $\mathrm{PA} \vdash \Box\bot \to \Box\psi$ for any ψ, so we have $\mathrm{PA} \vdash G \to \mathrm{Con}_{\mathrm{PA}}$. In the next section, we shall see that in fact, $\mathrm{PA} \vdash G \leftrightarrow \mathrm{Con}_{\mathrm{PA}}$, from which it follows that $\mathrm{PA} \not\vdash \mathrm{Con}_{\mathrm{PA}}$. This is Gödel's Second Incompleteness Theorem: PA does not prove its own consistency".

A number of exercises to finish this section:

Exercise 78 Show that for any formula $\varphi(v)$ with one free variable v, the set

$$\{n \in \mathbb{N} \,|\, \mathrm{PA} \vdash \varphi[\bar{n}/v]\}$$

is recursively enumerable. Conclude that if a function is numeralwise representable in PA, it is recursive, hence Σ_1-representable.

Exercise 79 Define a function $F : \mathbb{N} \to \mathbb{N}$ by:

$$F(n) = \max\{\mu m.\mathcal{N} \models \theta[n, j_0(m), j_1(m)] \,|\, \theta \in \Theta(n)\} + 1$$

where $\Theta(n)$ is the set of all Δ_0-formulas $\theta(u,v,w)$ such that

$$\ulcorner\theta(u,v,w)\urcorner < n \text{ and } \exists y < n\mathrm{Prf}(y, \ulcorner\forall u\exists v\exists w\theta(u,v,w)\urcorner)$$

(and the maximum of the empty set is 0).

i) Show that F is total recursive;

ii) show that F cannot be provably recursive.

Exercise 80 [Tarski's theorem on the non-definability of truth]. Apply Lemma 5.1 to show that there is no formula of $\mathcal{L}_{\mathrm{PA}}$ which defines the set of true $\mathcal{L}_{\mathrm{PA}}$-sentences, i.e. if

$$A = \{n \in \mathbb{N} \,|\, n \text{ is the code of a sentence } \varphi \text{ such that } \mathcal{N} \models \varphi\}$$

then there is no formula $\psi(v)$ such that for all $n \in \mathbb{N}$:

$$n \in A \Leftrightarrow \mathcal{N} \models \psi[n]$$

5.3 Gödel's Second Incompleteness Theorem

As we said in the preceding section, Gödel's Second Incompleteness Theorem asserts that "PA does not prove its own consistency". More formally: $\mathrm{PA} \not\vdash \mathrm{Con}_{\mathrm{PA}}$ (recall that $\mathrm{Con}_{\mathrm{PA}}$ is the sentence $\neg\Box\bot$).

Recall that we had derived (proposition 5.3) the following rules governing the operation $\Box$:

$$\begin{array}{ll} \mathrm{D1} & \mathrm{PA} \vdash \varphi \;\Rightarrow\; \mathrm{PA} \vdash \Box\varphi \\ \mathrm{D2} & \mathrm{PA} \vdash \Box(\varphi \to \psi) \wedge \Box\varphi \;\to \Box\psi \end{array}$$

Exercise 81 Prove that for *any* operation $\Box$, satisfying D1 anfd D2, one has:

$$\mathrm{PA} \vdash \Box(\varphi \wedge \psi) \leftrightarrow \Box\varphi \wedge \Box\psi$$

Our aim in this section is to prove that we have a third rule:

$$\begin{array}{ll} \mathrm{D3} & \mathrm{PA} \vdash \Box\varphi \to \Box\Box\varphi \end{array}$$

Let us see that this implies what we want:

Theorem 5.5 *For any operation $\Box$ satisfying* D1–D3 *and any G such that $\mathrm{PA} \vdash G \leftrightarrow \neg\Box G$, we have*

$$\mathrm{PA} \vdash G \leftrightarrow \neg\Box\bot$$

Proof. Since PA $\vdash \bot \to G$, by D1 and D2 we have PA $\vdash \Box\bot \to \Box G$, so PA $\vdash G \to \neg\Box G \to \neg\Box\bot$.

For the converse implication, we have from D2 and the assumption on G, PA $\vdash \Box G \to \Box(\neg\Box G)$; by D3 we have PA $\vdash \Box G \to \Box\Box G$. Combining the two, we have PA $\vdash \Box G \to \Box\bot$, so PA $\vdash \neg G \to \Box G \to \Box\bot$, whence PA $\vdash \neg\Box\bot \to G$. ∎

Corollary 5.6 (Gödel's Second Incompleteness Theorem)

$$\mathrm{PA} \nvdash \mathrm{Con}_{\mathrm{PA}}$$

Proof. Immediate. ∎

The rule D3, which we want to prove, is in fact a consequence of a more general theorem, which is known as "Formalized Σ_1-completeness". This is because $\Box\varphi$ is a Σ_1-sentence.

Theorem 5.7 (Formalized Σ_1-completeness of PA) *For every Σ_1-sentence of* PA*,*

$$\mathrm{PA} \vdash \varphi \to \Box\varphi$$

The rest of this section is devoted to the proof of theorem 5.7. Let us recall how we proved ordinary Σ_1-completeness. We proved that for any Δ_0-formula $\varphi(v_0, \ldots, v_{k-1})$ and for every k-tuple of natural numbers $n_0, \ldots, n_{k-1}$:

$$(\dagger) \qquad \mathcal{N} \models \varphi[n_0, \ldots, n_{k-1}] \;\Rightarrow\; \mathrm{PA} \vdash \varphi[\overline{n_0}/v_0, \ldots, \overline{n_{k-1}}/v_{k-1}]$$

We follow a similar line in the formalized case. We now assume that $\mathcal{L}_{\mathrm{PA}}$ is augmented with function symbols $\langle \cdot, \ldots, \cdot \rangle$, lh, $(\cdot)_i$ for the manipulation of sequences. We also take a function symbol T, representing the primitive recursive function $n \mapsto \ulcorner \overline{n} \urcorner$; and we want function symbols S_f and S_t representing the primitive recursive substitution operations on formulas and terms, respectively:

$$S_f(y,x) \quad = \quad \begin{cases} \ulcorner \varphi[s_0/v_0, \ldots, s_{k-1}/v_{k-1}] \urcorner & \text{if } y \text{ is a code for } \varphi, \\ & \mathrm{lh}(x) = k, \text{ and for each } i < k \\ & (x)_i \text{ is a code for } s_i \\ 0 & \text{else} \end{cases}$$

$$S_t(y,x) \quad = \quad \begin{cases} \ulcorner t[s_0/v_0, \ldots, s_{k-1}/v_{k-1}] \urcorner & \text{if } y \text{ is a code for } t, \\ & \mathrm{lh}(x) = k, \text{ and for each } i < k \\ & (x)_i \text{ is a code for } s_i \\ 0 & \text{else} \end{cases}$$

As before, we may assume that PA proves the recursions for these functions. In particular, we may assume that the sentences

$$\begin{aligned} T(0) &= \overline{\langle 0\rangle} \\ T(x+1) &= \langle \bar{3}, T(x), \overline{\langle 1\rangle}\rangle \\ S_t(\langle \bar{3}, \overline{\ulcorner t\urcorner}, \overline{\ulcorner s\urcorner}\rangle, x) &= \langle \bar{3}, S_t(\overline{\ulcorner t\urcorner}, x), S_t(\overline{\ulcorner s\urcorner}, x)\rangle \\ S_t(\langle \bar{4}, \overline{\ulcorner t\urcorner}, \overline{\ulcorner s\urcorner}\rangle, x) &= \langle \bar{4}, S_t(\overline{\ulcorner t\urcorner}, x), S_t(\overline{\ulcorner s\urcorner}, x)\rangle \\ S_f(\langle \bar{5}, \overline{\ulcorner t\urcorner}, \overline{\ulcorner s\urcorner}\rangle, x) &= \langle \bar{5}, S_t(\overline{\ulcorner t\urcorner}, x), S_t(\overline{\ulcorner s\urcorner}, x)\rangle \\ &\vdots \end{aligned}$$

are provable in PA. The formalization of statement (†) above is:

Lemma 5.8 *For every Δ_0-formula $\varphi(v_0, \ldots, v_{k-1})$ we have:*

$$\mathrm{PA} \vdash \forall x_0 \cdots x_{k-1}(\varphi(\vec{x}) \rightarrow \exists y \overline{\mathrm{Prf}}(y, S_f(\overline{\ulcorner \varphi \urcorner}, \langle T(x_0), \ldots, T(x_{k-1})\rangle)))$$

The proof of Lemma 5.8 goes via the auxiliary lemmas 5.9, 5.10 and 5.11 below.

Lemma 5.9

$$\begin{aligned} &\mathrm{PA} \vdash \forall xy \exists z \overline{\mathrm{Prf}}(z, \langle \bar{5}, T(x+y), S_t(\overline{\ulcorner v_0 + v_1 \urcorner}, \langle T(x), T(y)\rangle)\rangle) \\ &\mathrm{PA} \vdash \forall xy \exists z \overline{\mathrm{Prf}}(z, \langle \bar{5}, T(x \cdot y), S_t(\overline{\ulcorner v_0 \cdot v_1 \urcorner}, \langle T(x), T(y)\rangle)\rangle) \end{aligned}$$

Proof. Check, that these statements are formalizations of the statements that PA $\vdash \overline{n+m} = \overline{n} + \overline{m}$ and PA $\vdash \overline{n \cdot m} = \overline{n} \cdot \overline{m}$.

By the recursion equations for S_t we have that

$$S_t(\overline{\ulcorner v_0 + v_1 \urcorner}, \langle T(x), T(y)\rangle) = \langle \bar{3}, T(x), T(y)\rangle$$

so we must prove

$$\exists z \overline{\mathrm{Prf}}(z, \langle \bar{5}, T(x+y), \langle \bar{3}, T(x), T(y)\rangle\rangle)$$

which we do by induction on y. For $y = 0$, $T(y) = \overline{\langle 0\rangle}$ and we observe that

$$\langle \bar{5}, T(x), \langle \bar{3}, T(x), \overline{\langle 0\rangle}\rangle\rangle = S_f(\overline{\ulcorner v_0 = v_0 + 0 \urcorner}, \langle T(x)\rangle)$$

Since $\forall v_0(v_0 = v_0 + 0)$ is the universal closure of a PA-axiom, we have by one step (∀E),

$$\exists z \overline{\mathrm{Prf}}(z, S_f(\overline{\ulcorner v_0 = v_0 + 0 \urcorner}, \langle T(x)\rangle))$$

For the induction step, assume

$$\exists z \overline{\mathrm{Prf}}(z, \langle \bar{5}, T(x+y), \langle \bar{3}, T(x), T(y)\rangle\rangle)$$

Then by applying a substitution axiom for equality, also

$$\exists z\overline{\mathrm{Prf}}(z, \langle\bar{5}, \langle\bar{3}, T(x+y), \overline{\langle 1\rangle}\rangle, \langle\bar{3}, \langle\bar{3}, T(x), T(y)\rangle, \overline{\langle 1\rangle}\rangle\rangle)$$

By an application of the axiom $\forall uv((u+v)+1 = u+(v+1))$ we have

$$\exists z\overline{\mathrm{Prf}}(z, \langle\bar{5}, \langle\bar{3}, \langle\bar{3}, T(x), T(y)\rangle, \overline{\langle 1\rangle}\rangle, \langle\bar{3}, T(x), \langle\bar{3}, T(y), \overline{\langle 1\rangle}\rangle\rangle\rangle)$$

But $\langle\bar{3}, T(y), \overline{\langle 1\rangle}\rangle = T(y+1)$ by the recursion equations for T, which also give $\langle\bar{3}, T(x+y), \overline{\langle 1\rangle}\rangle = T(x+(y+1)) = T((x+y)+1)$, so by applying transitivity of equality we get

$$\exists z\overline{\mathrm{Prf}}(z, \langle\bar{5}, T(x+(y+1)), \langle\bar{3}, T(x), T(y+1)\rangle\rangle)$$

as desired.

The proof of the second statement is similar (and uses the first!). ∎

The proof of lemma 5.9 was, of course, quite unreadable, but the point is that one has a precise idea of what one is doing. One cannot write, for example, that $\langle\bar{3}, T(x), T(y)\rangle = \ulcorner T(x)+T(y)\urcorner$; but, $T(x)$ and $T(y)$ are, "in PA", codes for *terms* $\tilde{x}$ and $\tilde{y}$, so that "$\langle\bar{3}, T(x), T(y)\rangle = \ulcorner\tilde{x}+\tilde{y}\urcorner$" but again this is imprecise, because our coding acts on real terms only. The following notational convention gives a precise way of getting some clarification: for any formula $\varphi(v_0, \ldots, v_{k-1})$, we let

$$\ulcorner\varphi(\widetilde{x_0}, \ldots, \widetilde{x_{k-1}})\urcorner$$

be an abbreviation for $S_f(\overline{\ulcorner\varphi\urcorner}, \langle T(x_0), \ldots, T(x_{k-1})\rangle)$. We write

$$\boxdot\varphi(\widetilde{x_0}, \ldots, \widetilde{x_{k-1}})$$

for $\exists z\overline{\mathrm{Prf}}(z, \ulcorner\varphi(\widetilde{x_0}, \ldots, \widetilde{x_{k-1}})\urcorner)$. With these conventions, Lemma 5.9 becomes:

$$\begin{array}{l} \mathrm{PA} \vdash \forall xy \boxdot (\widetilde{x+y} = \widetilde{x} + \widetilde{y}) \\ \mathrm{PA} \vdash \forall xy \boxdot (\widetilde{x\cdot y} = \widetilde{x}\cdot\widetilde{y}) \end{array}$$

It is now straightforward (by induction on the term) to show that for any term $t(v_0, \ldots, v_{k-1})$ we have:

$$\mathrm{PA} \vdash \forall x_0 \cdots x_{k-1} \boxdot \widetilde{t(x_0, \ldots, x_{k-1})} = t(\widetilde{x_0}, \ldots, \widetilde{x_{k-1}})$$

Exercise 82 Carry out this proof.

The following lemma is an immediate consequence.

Lemma 5.10 *For terms $t(v_0,\ldots,v_{k-1})$ and $s(v_0,\ldots,v_{k-1})$ we have*

$$\mathrm{PA} \vdash \forall x_0 \cdots x_{k-1}(t(\vec{x}) = s(\vec{x}) \to \boxdot(t(\widetilde{x_0},\ldots,\widetilde{x_{k-1}}) = s(\widetilde{x_0},\ldots,\widetilde{x_{k-1}})))$$
$$\mathrm{PA} \vdash \forall x_0 \cdots x_{k-1}(t(\vec{x}) < s(\vec{x}) \to \boxdot(t(\widetilde{x_0},\ldots,\widetilde{x_{k-1}}) < s(\widetilde{x_0},\ldots,\widetilde{x_{k-1}})))$$

We are now ready for the final induction.

Lemma 5.11 *Let Φ be the set of formulas $\varphi(v_0,\ldots,v_{k-1})$ for which*

$$\mathrm{PA} \vdash \forall x_0 \cdots x_{k-1}(\varphi(x_0,\ldots,x_{k-1}) \to \boxdot\varphi(\widetilde{x_0},\ldots,\widetilde{x_{k-1}}))$$

Then Φ contains all formulas of form $t = s$ and $t < s$, and Φ is closed under conjunction, disjunction and bounded quantification.

Proof. That Φ contains all formulas $t = s$ and $t < s$, is lemma 5.10. The induction steps for $\wedge$ and $\vee$ are easy.

Now suppose $\varphi(v_0,\ldots,v_{k-1})$ has the form $\exists v_k < v_0\psi(v_0,\ldots,v_k)$, for $\psi \in \Phi$. Then $\forall x_0 \cdots x_{k-1}(\varphi(\vec{x}) \to \boxdot\varphi(\widetilde{x_0},\ldots,\widetilde{x_{k-1}}))$ is equivalent (in PA) to

$$\forall x_0 \cdots x_k(x_k < x_0 \wedge \psi(x_0,\ldots,x_k) \to \boxdot(\exists v_k < \widetilde{x_0}\,\psi(\widetilde{x_0},\ldots,\widetilde{x_{k-1}},v_k)))$$

Since $\psi \in \Phi$, $v_k < v_0 \in \Phi$ and by the induction step for $\wedge$, we have

$$\mathrm{PA} \vdash \forall x_0 \cdots x_k(x_k < x_0 \wedge \psi(x_0,\ldots,x_k) \to \boxdot(\widetilde{x_k} < \widetilde{x_0} \wedge \psi(\widetilde{x_0},\ldots,\widetilde{x_k})))$$

so the desired conclusion follows by an application of $\exists$I.
Now suppose φ is $\forall v_k < v_0\psi(v_0,\ldots,v_k)$ with $\psi \in \Phi$. We prove the implication:

$$\forall v_k < x_0\psi(x_0,\ldots,x_{k-1},v_k) \to \boxdot(\forall v_k < \widetilde{x_0}\psi(\widetilde{x_0},\ldots,\widetilde{x_{k-1}},v_k))$$

by induction on x_0. For x_0 it holds trivially; for the induction step we observe that

$$\forall v_k < x_0 + 1\psi \leftrightarrow \forall v_k < x_0\psi \wedge \psi(x_0,\ldots,x_{k-1},x_0)$$

so that

$$\forall v_k < v_0\psi \to \boxdot(\forall v_k < \widetilde{x_0}\psi(\widetilde{x_0},\ldots,\widetilde{x_{k-1}},v_k) \wedge \psi(\widetilde{x_0},\ldots,\widetilde{x_{k-1}},\widetilde{x_0}))$$

We also have $\forall x_0 \boxdot (\widetilde{x_0 + 1} = \widetilde{x_0} + 1)$ and

$$\forall x_0 \boxdot (\forall v_k(v_k < \widetilde{x_0} + 1 \leftrightarrow v_k < \widetilde{x_0} \vee v_k = \widetilde{x_0}))$$

so we obtain the desired implication

$$\forall v_k < x_0 + 1\psi \to \boxdot\forall v_k < \widetilde{x_0}\psi(\widetilde{x_0},\ldots,\widetilde{x_{k-1}},v_k)$$

$\blacksquare$

Exercise 83 i) Show that lemma 5.11 is sufficient to prove Lemma 5.8. That is, show that the set Φ contains all Δ_0-formulas;

ii) show that, in turn, Lemma 5.8 implies Theorem 5.7.

Remark The proof of Gödel's Incompleteness Theorems can be carried out for any recursively enumerable extension of PA. By this we mean: a theory, formulated in a language which is coded in a recursive way, and with axioms whose codes form an r.e. set. In fact, we don't need the full force of PA here. Any recursively enumerable theory T which has enough arithmetic to represent (and prove the recursion equations of) the necessary primitive recursive functions, can formulate its own consistency Con_T, and if T is consistent, then $T \not\vdash \mathrm{Con}_T$.

An important example is ZFC: set theory with the axiom of Choice. Here is an example of an application of Gödel's Second Incompleteness Theorem to ZFC. A cardinal number κ is called *strongly inaccessible* if $\kappa > \aleph_0$, κ is regular, and $\forall \lambda < \kappa(2^\lambda < \kappa)$. One can prove, in ZFC, that if κ is strongly inaccessible, then V_κ is a model of ZFC. Therefore, in ZFC, if κ is strongly inaccessible, ZFC is consistent. By Gödel's Second Incompleteness Theorem, ZFC $\not\vdash$ I where I is the statement: there is a strongly inaccessible cardinal. But one may wish to know whether ZFC+I is consistent. The question becomes: assuming $\mathrm{Con}_{\mathrm{ZFC}}$, can we prove $\mathrm{Con}_{\mathrm{ZFC+I}}$? Again *no*, for we have seen that $\mathrm{ZFC} + \mathrm{I} \vdash \mathrm{Con}_{\mathrm{ZFC}}$, so if $\mathrm{ZFC} + \mathrm{Con}_{\mathrm{ZFC}} \vdash \mathrm{Con}_{\mathrm{ZFC+I}}$, then $\mathrm{ZFC} + \mathrm{I} \vdash \mathrm{Con}_{\mathrm{ZFC+I}}$ which contradicts the Second Incompleteness Theorem, applied to the theory ZFC+I.

Another application of Theorem 5.6 to an extension of PA is *Löb's Theorem.* Löb's theorem says that although the formula $\Box\varphi \to \varphi$ is true in $\mathcal{N}$, it is provable in PA only if φ is provable in PA:

Theorem 5.12 (Löb's Theorem) *If* $\mathrm{PA} \vdash \Box\varphi \to \varphi$, *then* $\mathrm{PA} \vdash \varphi$.

Proof. If $\mathrm{PA} \not\vdash \varphi$ then $\mathrm{PA} + \neg\varphi$ is consistent, so by the Second Incompleteness Theorem, applied to $\mathrm{PA} + \neg\varphi$, $\mathrm{PA} + \neg\varphi \not\vdash \mathrm{Con}_{\mathrm{PA}+\neg\varphi}$. But now, in PA, $\mathrm{Con}_{\mathrm{PA}+\neg\varphi}$ is equivalent to $\neg\Box\varphi$. So we have $\mathrm{PA} + \neg\varphi \not\vdash \neg\Box\varphi$, whence $\mathrm{PA} \not\vdash \Box\varphi \to \varphi$. ∎

Exercise 84 Prove Löb's Theorem directly from Lemma 5.1, by taking a sentence ψ such that

$$\mathrm{PA} \vdash \psi \leftrightarrow \Box(\psi \to \varphi)$$

Use the properties D1–D3.

Exercise 85 As before, but now take ψ satisfying

$$\mathrm{PA} \vdash \psi \leftrightarrow (\Box\psi \rightarrow \varphi)$$

Chapter 6

Introduction to Models of PA

6.1 The theory PA^- and end-extensions

From now on, we take the symbol $<$ as part of the language $\mathcal{L}_{\mathrm{PA}}$, so every $\mathcal{L}_{\mathrm{PA}}$-structure $\mathcal{M}$ carries a binary relation $<^{\mathcal{M}}$.

The symbol $\mathcal{N}$ will *always* denote the standard model.

We shall find it useful to consider some $\mathcal{L}_{\mathrm{PA}}$-structures that are not models of PA, but of a weaker theory PA^-, which we therefore now introduce.

Definition 6.1 PA^- is the $\{+, \cdot; <; 0, 1\}$-theory with axioms stating that:

1) $+$ and $\cdot$ are commutative and associative and $\cdot$ distributes over $+$;

2) $\forall x(x{\cdot}0 = 0 \wedge x{\cdot}1 = x \wedge x + 0 = x)$

3) $<$ is a linear order satisfying $\forall x(0 \leq x)$ and $\forall x(0 < x \leftrightarrow 1 \leq x)$

4) $\forall xyz(x < y \rightarrow x + z < y + z)$

5) $\forall xyz(0 < z \wedge x < y \rightarrow x{\cdot}z < y{\cdot}z)$

6) $\forall xy(x < y \rightarrow \exists z(x + z = y))$

So, every model of PA^- is a linear order. If $\mathcal{M}_1$ and $\mathcal{M}_2$ are $\mathcal{L}_{\mathrm{PA}}$-structures and $\mathcal{M}_1$ is a substructure of $\mathcal{M}_2$, we say that $\mathcal{M}_1$ is an *initial segment* of $\mathcal{M}_2$, or $\mathcal{M}_2$ is an *end-extension* of $\mathcal{M}_1$, if for all $m \in \mathcal{M}_2$ and $n \in \mathcal{M}_1$, if $\mathcal{M}_2 \models m < n$ then $m \in \mathcal{M}_1$. Notation: $\mathcal{M}_1 \subseteq_e \mathcal{M}_2$ (see also definition 6.7 below).

If $\mathcal{M}$ is any model of PA$^-$, the function $n \mapsto \overline{n}^{\mathcal{M}} : \mathbb{N} \to \mathcal{M}$ is an embedding of $\mathcal{L}_{\mathrm{PA}}$-structures.

Exercise 86 Prove this, and prove also that this mapping embeds $\mathcal{N}$ as initial segment in $\mathcal{M}$.

If Γ is a set of $\mathcal{L}_{\mathrm{PA}}$-formulas, and $\mathcal{M}_1$ an $\mathcal{L}_{\mathrm{PA}}$-substructure of $\mathcal{M}_2$, we say that $\mathcal{M}_1$ is a Γ-*elementary* substructure of $\mathcal{M}_2$, notation: $\mathcal{M}_1 \prec_\Gamma \mathcal{M}_2$, if for every $\varphi(v_1, \ldots, v_k) \in \Gamma$ and all k-tuples $m_1, \ldots, m_k \in \mathcal{M}_1$,

$$\mathcal{M}_1 \models \varphi[m_1, \ldots, m_k] \Leftrightarrow \mathcal{M}_2 \models \varphi[m_1, \ldots, m_k]$$

We also say that $\mathcal{M}_2$ is a Γ-*elementary extension* of $\mathcal{M}_1$. If Γ is the set of *all* $\mathcal{L}_{\mathrm{PA}}$-formulas, we drop Γ in the notation and speak of "elementary substructure/extension".

Exercise 87 Let $\mathcal{M}_1 \subseteq_e \mathcal{M}_2$ and $\mathcal{M}_1$, $\mathcal{M}_2$ models of PA$^-$. Show that $\mathcal{M}_1 \prec_{\Delta_0} \mathcal{M}_2$.

A useful criterion for testing whether an inclusion of models of PA is an elementary extension, is the 'Tarski-Vaught test', given below as an exercise.

Exercise 88 [Tarski-Vaught test] Suppose $\mathcal{M}_1$ is an $\mathcal{L}_{\mathrm{PA}}$-substructure of $\mathcal{M}_2$. Then it is an elementary substructure if and only if for every formula $\varphi(x_1, \ldots, x_k, y)$ and every k-tuple $a_1, \ldots, a_k$ of elements of $\mathcal{M}_1$, the following holds: if $\mathcal{M}_2 \models \exists y \varphi(a_1, \ldots, a_k, y)$ then there exists a $c \in \mathcal{M}_1$ such that $\mathcal{M}_2 \models \varphi(a_1, \ldots, a_k, c)$.

Exercise 89 Show that for any inclusion $\mathcal{M}_1 \subseteq \mathcal{M}_2$ of models of PA, that $\mathcal{M}_1 \prec_{\Delta_0} \mathcal{M}_2$ implies $\mathcal{M}_1 \prec_{\Delta_1} \mathcal{M}_2$.

Exercise 90 Show that PA$^-$ proves all true Σ_1-sentences.

Exercise 91 Show that for $\mathcal{L}_{\mathrm{PA}}$-structures $\mathcal{M}_1$ and $\mathcal{M}_2$: if $\mathcal{M}_1 \subseteq_e \mathcal{M}_2$ and $\mathcal{M}_2$ is a model of PA$^-$, then $\mathcal{M}_1$ is a model of PA$^-$.

6.2 Cuts, Overspill and Underspill

Let $\mathcal{M}$ be a model of PA. A *cut* of $\mathcal{M}$ is a nonempty subset $I \subseteq \mathcal{M}$ such that $x < y$ and $y \in I$ implies $x \in I$, and $x \in I$ implies $x + 1 \in I$. A cut I is *proper* if $I \neq \mathcal{M}$. The following easy lemma is of fundamental importance in the study of nonstandard models of PA.

Lemma 6.2 *Let $\mathcal{M}$ be a model of* PA*, and $I \subset \mathcal{M}$ a proper cut. Then I is not definable in parameters from $\mathcal{M}$, that is: there is no $\mathcal{L}_{\mathrm{PA}}$-formula $\varphi(v_1, \ldots, v_{k+1})$ such that for some $m_1, \ldots, m_k \in \mathcal{M}$:*

$$I = \{m \in \mathcal{M} \,|\, \mathcal{M} \models \varphi[m_1, \ldots, m_k, m]\}$$

Proof. Since I is nonempty, $0 \in I$. Moreover, $m \in I$ implies $m + 1 \in I$. Were I definable by φ in parameters $m_1, \ldots, m_k$ as in the Lemma, then since $\mathcal{M}$ satisfies induction, we would have $I = \mathcal{M}$. ∎

Corollary 6.3 (Overspill Lemma) *Let $\mathcal{M}$ be a model of* PA *and $I \subset \mathcal{M}$ a proper cut. If $m_1, \ldots, m_k \in \mathcal{M}$ and $\mathcal{M} \models \varphi[m_1, \ldots, m_k, b]$ for every $b \in I$, then there is $c \in \mathcal{M} \setminus I$ such that*

$$\mathcal{M} \models \forall y \leq c \varphi[m_1, \ldots, m_k, y]$$

Proof. Certainly, for all $c \in I$ we have $\mathcal{M} \models \forall y \leq c\varphi[m_1, \ldots, m_k, y]$; so if such $c \in \mathcal{M} \setminus I$ would not exist, we would have

$$I = \{c \,|\, \mathcal{M} \models \forall y \leq c\varphi[m_1, \ldots, m_k, y]\}$$

contradicting the non-definability of I of Lemma 6.2. ∎

Corollary 6.4 *Again let $\mathcal{M}$ be a model of* PA *and $I \subset \mathcal{M}$ a proper cut. Suppose that for φ, $m_1, \ldots, m_k \in \mathcal{M}$ we have: for all $x \in I$ there is $y \in I$ with*

$$\mathcal{M} \models y \geq x \wedge \varphi[m_1, \ldots, m_k, y]$$

Then for each $c \in \mathcal{M} \setminus I$ there is $b \in \mathcal{M} \setminus I$ with

$$\mathcal{M} \models b < c \wedge \varphi[m_1, \ldots, m_k, b]$$

Proof. Apply Corollary 6.3 to the formula

$$\exists y(x \leq y < c \wedge \varphi[m_1, \ldots, m_k, y])$$

∎

Corollary 6.5 (Underspill Lemma) *Let $\mathcal{M}$ a model of* PA *and $I \subset \mathcal{M}$ a proper cut.*

i) If for all $c \in \mathcal{M} \setminus I$, $\mathcal{M} \models \varphi[m_1, \ldots, m_k, c]$, then there is $b \in I$ such that $\mathcal{M} \models \forall x \geq b \varphi[m_1, \ldots, m_k, x]$;

ii) if for all $c \in \mathcal{M} \backslash I$ there is $x \in \mathcal{M} \backslash I$ with $\mathcal{M} \models x < c \wedge \varphi[m_1, \ldots, m_k, x]$, then for all $b \in I$ there is $y \in I$ with $\mathcal{M} \models b < y \wedge \varphi[m_1, \ldots, m_k, y]$.

Exercise 92 Prove Corollary 6.5.

6.3 The ordered Structure of Models of PA

We study now the order-type of models of PA; that is, their $\{<\}$-reduct.

If A and B are two linear orders, we order the set $A \times B$ lexicographically, that is: $(a, b) < (a', b')$ iff $a < a'$ or $a = a' \wedge b < b'$. $A \times B$ is then also a linear order, and the picture is: replace every $a \in A$ by a copy of B. By $A + B$ we mean the ordered set which is the disjoint union of A and B, and in which every element of A is below every element of B.

Theorem 6.6 *Let $\mathcal{M}$ be a nonstandard model of* PA*. Then as ordered set, $\mathcal{M} \cong \mathbb{N} + A \times \mathbb{Z}$ where A is a dense, linear order without end-points. Therefore, if $\mathcal{M}$ is countable, $\mathcal{M} \cong \mathbb{N} + \mathbb{Q} \times \mathbb{Z}$*

Proof. $\mathcal{M}$ has $\mathcal{N}$ as initial segment, so $\mathcal{M} \cong \mathbb{N} + X$ for some linear order X. For nonstandard $a \in \mathcal{M}$, let $Z(a)$ the set of elements of $\mathcal{M}$ which differ from a by a standard element: $a' \in Z(a)$ iff for some $n \in \mathbb{N}$, $\mathcal{M} \models a' + \overline{n} = a \vee a + \overline{n} = a'$. If $a, b \in \mathcal{M}$ are nonstandard elements and $a \notin Z(b)$, then $Z(a) \cap Z(b) = \emptyset$, and if moreover $a < b$, we have $x < y$ for every $x \in Z(a)$ and $y \in Z(b)$. Since clearly, every $Z(a)$ is order-isomorphic to $\mathbb{Z}$, we have $\mathcal{M} \cong \mathbb{N} + A \times \mathbb{Z}$, where A is the collection of all sets $Z(a)$, ordered by: $Z(a) < Z(b)$ iff $a < b$.

Now A is dense, for given a, b nonstandard, if $Z(a) < Z(b)$ then $Z(a) < Z([\frac{a+b}{2}]) < Z(b)$ (check!).

A has no endpoints: for every nonstandard a we have $Z([\frac{a}{2}]) < Z(a) < Z(a + a)$ (check this too!).

The final statement of the theorem follows from the well-known fact that every countable dense linear order without end-points is order-isomorphic to $\mathbb{Q}$. ∎

We shall now see some examples of proper cuts of a nonstandard model $\mathcal{M}$.

Definition 6.7 An *initial segment* of an $\mathcal{L}_{\text{PA}}$-structure $\mathcal{M}$ is a cut which is closed under the operations $+, \cdot$ in $\mathcal{M}$ (such cuts are then $\mathcal{L}_{\text{PA}}$-substructures of $\mathcal{M}$, and hence models of PA^-, if $\mathcal{M}$ is).

Examples.

1) Let $\mathcal{M}$ be a nonstandard model of PA, and $a \in \mathcal{M}$ nonstandard. By $a^{\mathbb{N}}$ we mean the set

$$\{m \in \mathcal{M} \,|\, \text{for some } n \in \mathbb{N},\, \mathcal{M} \models m < a^n\}$$

Convince yourself that $a^{\mathbb{N}}$ is closed under the operations $+, \cdot$ of $\mathcal{M}$. Moreover, $a \in a^{\mathbb{N}}$. It is easy to see, that $a^{\mathbb{N}}$ is the smallest initial segment of $\mathcal{M}$ that contains a. It is also easy to see, that $a^{\mathbb{N}} \neq \mathcal{M}$, for $a^a \notin a^{\mathbb{N}}$. By the same token, $a^{\mathbb{N}}$ is not a model of PA.

2) Let $a \in \mathcal{M}$ be nonstandard as before. By $a^{1/\mathbb{N}}$ we mean the set

$$\{m \in \mathcal{M} \,|\, \text{for all } n \in \mathbb{N},\, \mathcal{M} \models m^n < a\}$$

Again, $a^{1/\mathbb{N}}$ is closed under $+, \cdot$ and is a proper initial segment since $a \notin a^{1/\mathbb{N}}$. Since $\mathbb{N} \subseteq a^{1/\mathbb{N}}$, for every $n \in \mathbb{N}$ we have $\mathcal{M} \models n^n < a$; by the Overspill Lemma, there is a nonstandard element $c \in \mathcal{M}$ such that $\mathcal{M} \models c^c < a$. Clearly then, $c \in a^{1/\mathbb{N}} \setminus \mathbb{N}$.

The following exercises both require use of the Overspill Lemma.

Exercise 93 Show that for $a \in \mathcal{M}$ nonstandard, $m \in \mathcal{M} \setminus a^{\mathbb{N}}$ if and only if $a^c < m$ for some nonstandard $c \in \mathcal{M}$.

Exercise 94 Let $a \in \mathcal{M}$ be nonstandard.

a) Show that for each $n \in \mathbb{N}$ there is $b \in \mathcal{M}$ such that $\mathcal{M} \models b^n \leq a < (b+1)^{n+1}$. Show that for each such b, $\mathcal{M} \models b^b > a$;

b) show that $a^{1/\mathbb{N}}$ is not a model of PA, by showing that there is $c \in a^{1/\mathbb{N}}$ with $\mathcal{M} \models c^c > a$.

The following exercise explains the name “cut”.

Exercise 95 Let $\mathcal{M}$ be a countable nonstandard model of PA and $I \subseteq \mathcal{M}$ a proper cut which is not the standard cut $\mathbb{N}$. Suppose that I is closed under $+$. Then under the identification $\mathcal{M} \cong \mathbb{N} + \mathbb{Q} \times \mathbb{Z}$ of 6.6, I corresponds to $\mathbb{N} + A \times \mathbb{Z}$, where $A \subset \mathbb{Q}$ is a *Dedekind cut*: a set of form $\{q \in \mathbb{Q} \,|\, q < r\}$ for some real number r.

Exercise 96 Let $\mathcal{M}$ be a nonstandard model of PA; by theorem 6.6, write $\mathcal{M} \cong \mathbb{N} + A \times \mathbb{Z}$ as ordered structures, with A a dense linear order without end-points. Show that A cannot be order-isomorphic to the real line $\mathbb{R}$ [Hint: let $m \in \mathcal{M}$ be nonstandard and consider the set $\{Z(m{\cdot}\bar{n}) \,|\, n \in \mathbb{N}\}$ as subset of A].

Theorem 6.8 *Let $\mathcal{M}$ be a countable, nonstandard model of* PA. *Then $\mathcal{M}$ has $2^{\aleph_0}$ proper cuts which are closed under $+$ and $\cdot$.*

Proof. Define an equivalence relation on the set of nonstandard elements of $\mathcal{M}$ by: $a \sim b$ iff for some $n \in \mathbb{N}$,

$$a \leq b < a^n \text{ or } b \leq a < b^n$$

Clearly, this is an equivalence relation, and the set A of $\sim$-equivalence classes of $\mathcal{M} \setminus \mathbb{N}$ is linearly ordered by $[a] <_A [b]$ iff $a < b$ in $\mathcal{M}$. Suppose $[a] <_A [b]$. Then $a^n < b$ for each $n \in \mathbb{N}$. So for each $n \in \mathbb{N}$, there is x with $a^n < x < x^{n+2} < b$; that is, the formula

$$\exists x (a^y < x < x^{y+2} < b)$$

is satisfied (in $\mathcal{M}$) by all standard elements y. By the Overspill Lemma, there is a nonstandard c such that for some $x \in \mathcal{M}$,

$$a^c < x < x^c < b$$

It follows that $[a] <_A [x] <_A [b]$. So the ordering $(A, <_A)$ is dense, and by a similar overspill argument one deduces that it has no end points.

Therefore, since $\mathcal{M}$ was countable, there is an isomorphism $(A, <_A) \cong (\mathbb{Q}, <)$ and hence a surjective, $\leq$-preserving map

$$\mathcal{M} \setminus \mathbb{N} \to (\mathbb{Q}, <)$$

The inverse image of each Dedekind cut in $\mathbb{Q}$ defines a proper cut in $\mathcal{M}$, which is closed under $+$ and $\cdot$. Since there are $2^{\aleph_0}$ Dedekind cuts in $\mathbb{Q}$, the theorem is proved. ∎

6.4 MRDP Theorem and Gaifman's Splitting Theorem

Initial segments are one extreme of inclusions of models; the other extreme is the notion of a *cofinal* submodel. If $\mathcal{M}_1 \subseteq \mathcal{M}_2$ are models of PA^-, we say that $\mathcal{M}_1$ is cofinal in $\mathcal{M}_2$, or $\mathcal{M}_2$ is a *cofinal extension* of $\mathcal{M}_1$, if for every $m \in \mathcal{M}_2$ there is $m' \in \mathcal{M}_1$ such that $m < m'$ in $\mathcal{M}_2$. Notation: $\mathcal{M}_1 \subseteq_{\mathrm{cf}} \mathcal{M}_2$.

We extend the notions of Σ_1 and Π_1-formulas to arbitrary n, by putting inductively: a formula is Σ_{n+1} iff it is of form $\exists\vec{y}\psi$ with $\psi \in \Pi_n$; a formula is Π_{n+1} iff it is of form $\forall\vec{y}\psi$ with $\psi \in \Sigma_n$. Clearly, every formula is (up to equivalence in predicate logic) Σ_n for some n. In the definition of Σ_n and Π_n we allow the string $\vec{y}$ to be empty, so that every Σ_n-formula is automatically Σ_{n+1} and Π_{n+1}. A formula which is both Σ_n and Π_n is called a Δ_n-formula.

First an easy lemma which gives a simplified condition for when an extension is Σ_n-elementary.

Lemma 6.9 *Let $\mathcal{M}_1 \subseteq \mathcal{M}_2$ be an inclusion of $\mathcal{L}_{\mathrm{PA}}$-structures. If $n > 0$ and for each Σ_n-formula $\theta(\vec{x})$ and every tuple $\vec{a}$ of elements of $\mathcal{M}_1$ we have*

$$\mathcal{M}_2 \models \theta[\vec{a}] \Rightarrow \mathcal{M}_1 \models \theta[\vec{a}]$$

then $\mathcal{M}_1 \prec_{\Sigma_n} \mathcal{M}_2$.

Proof. For the converse direction, let $\theta(\vec{x}) \equiv \exists\vec{y}\varphi(\vec{x},\vec{y})$ (with $\varphi \in \Pi_{n-1}$) and suppose $\mathcal{M}_1 \models \theta[\vec{a}]$ so $\mathcal{M}_1 \models \varphi[\vec{a},\vec{b}]$ for some tuple $\vec{b}$ of elements of $\mathcal{M}_1$. Since $\neg\varphi$ is trivially Σ_n, we cannot have $\mathcal{M}_2 \models \neg\varphi[\vec{a},\vec{b}]$; so $\mathcal{M}_2 \models \varphi[\vec{a},\vec{b}]$ hence $\mathcal{M}_2 \models \theta[\vec{a}]$. ■

Theorem 6.10 *Let $\mathcal{M}_1 \subseteq_{\mathrm{cf}} \mathcal{M}_2$ be a cofinal extension of models of* PA^- *such that $\mathcal{M}_1 \prec_{\Delta_0} \mathcal{M}_2$. If $\mathcal{M}_1$ is a model of* PA *then $\mathcal{M}_1 \prec \mathcal{M}_2$.*

Proof. First we prove, using the criterion of lemma 6.9, that $\mathcal{M}_1 \prec_{\Sigma_2} \mathcal{M}_2$; and then that for $n \geq 2$, if $\mathcal{M}_1 \prec_{\Sigma_n} \mathcal{M}_2$ then $\mathcal{M}_1 \prec_{\Sigma_{n+1}} \mathcal{M}_2$.

Let $\theta(\vec{x})$ be a Σ_2-formula, $\theta(\vec{x}) \equiv \exists\vec{y}\forall\vec{z}\psi(\vec{x},\vec{y},\vec{z})$ with $\psi \in \Delta_0$, and suppose for $\vec{a} \in \mathcal{M}_1$ that $\mathcal{M}_2 \models \theta[\vec{a}]$, so there is $\vec{b} = b_1,\ldots,b_k$ in $\mathcal{M}_2$ such that $\mathcal{M}_2 \models \forall\vec{z}\psi[\vec{a},\vec{b},\vec{z}]$. Now $\mathcal{M}_1 \subseteq_{\mathrm{cf}} \mathcal{M}_2$, so there is $b \in \mathcal{M}_1$ with $b_1,\ldots,b_k < b$; then $\mathcal{M}_2 \models \exists\vec{y} < b\forall\vec{z}\psi[\vec{a},\vec{y},\vec{z}]$. Then certainly for all $c \in \mathcal{M}_1$ we have

$$\mathcal{M}_2 \models \exists\vec{y} < b\forall\vec{z} < c\psi[\vec{a},\vec{y},\vec{z}]$$

This is a Δ_0-formula, so because $\mathcal{M}_1 \prec_{\Delta_0} \mathcal{M}_2$ we have

$$\mathcal{M}_1 \models \forall w\exists\vec{y} < b\forall\vec{z} < w\psi[\vec{a},\vec{y},\vec{z}]$$

Now we use the assumption that $\mathcal{M}_1$ is a model of PA and satisfies therefore the Collection Principle: it follows, that

$$\mathcal{M}_1 \models \exists \vec{y} < b \forall \vec{z} \psi[\vec{a}, \vec{y}, \vec{z}]$$

(since its negation $\forall \vec{y} < b \exists \vec{z} \neg \psi$ implies, by Collection, $\exists w \forall \vec{y} < b \exists \vec{z} < w \neg \psi$) In particular, $\mathcal{M}_1 \models \exists \vec{y} \forall \vec{z} \psi[\vec{a}, \vec{y}, \vec{z}]$. By lemma 6.9 we may conclude that $\mathcal{M}_1 \prec_{\Sigma_2} \mathcal{M}_2$.

For the inductive step, now assume $\mathcal{M}_1 \prec_{\Sigma_n} \mathcal{M}_2$ for $n \geq 2$. Then since $\mathcal{M}_1$ is a model of PA and $\mathcal{M}_1 \prec_{\Sigma_2} \mathcal{M}_2$, the pairing function is a bijection from $\mathcal{M}_2^2$ to $\mathcal{M}_2$ (because this is expressed by a Π_2-formula which is true in $\mathcal{M}_1$). This has for effect that we can contract strings of quantifiers into single quantifiers, so for a Π_{n+1}-formula $\psi(\vec{x})$ we may assume it has the form $\psi \equiv \forall y \exists z \varphi(\vec{x}, y, z)$ with $\varphi \in \Pi_{n-1}$.

Suppose for $\vec{a} \in \mathcal{M}_1$ that $\mathcal{M}_1 \models \psi[\vec{a}]$. In order to show $\mathcal{M}_2 \models \psi[\vec{a}]$, we show that for each $b \in \mathcal{M}_1$, $\mathcal{M}_2 \models \forall y < b \exists z \varphi[\vec{a}, y, z]$, which suffices since $\mathcal{M}_1 \subseteq_{\mathrm{cf}} \mathcal{M}_2$.

Recall Theorem 4.9; since $\mathcal{M}_1 \models \forall y \exists z \varphi$ and $\mathcal{M}_1$ is a model of PA, by the induction axioms of PA we have

$$\mathcal{M}_1 \models \exists a, m \forall y < b \forall z (z = (a, m)_y \rightarrow \varphi[\vec{a}, y, z])$$

But this is Σ_n (check!), so

$$\mathcal{M}_2 \models \exists a, m \forall y < b \forall z (z = (a, m)_y \rightarrow \varphi[\vec{a}, y, z])$$

Since certainly $\mathcal{M}_2 \models \forall a, m \forall y \exists z (z = (a, m)_y)$ (because this is a Π_2-formula), we have that $\mathcal{M}_2 \models \forall y < b \exists z \varphi[\vec{a}, y, z]$, as desired.

We have proved: $\mathcal{M}_1 \models \psi[\vec{a}] \Rightarrow \mathcal{M}_2 \models \psi[\vec{a}]$ for every Π_{n+1}-formula $\psi(\vec{x})$ and every tuple $\vec{a}$ from $\mathcal{M}_1$; so $\mathcal{M}_2 \models \psi[\vec{a}] \Rightarrow \mathcal{M}_1 \models \psi[\vec{a}]$ for every Σ_{n+1}-formula $\psi(\vec{x})$ and every tuple $\vec{a}$ from $\mathcal{M}_1$; by lemma 6.9, we are done. ∎

The following result we need, although very easy to state, is quite deep, and we won't prove it. It is the famous Matiyasevich-Robinson-Davis-Putnam Theorem, which was used to show that Hilbert's 10th Problem cannot be solved (there is no algorithm which decides for an arbitrary polynomial $P(\vec{x})$ with integer coefficients and an arbitrary number of unknowns, whether the equation $P(\vec{x}) = 0$ has a solution in the integers).

Theorem 6.11 (MRDP Theorem) *For every Σ_1-formula $\varphi(\vec{x})$ there is a formula $\psi(\vec{x})$ of form $\exists \vec{y} \chi(\vec{x}, \vec{y})$ with χ quantifier-free, such that*

$$\mathrm{PA} \vdash \forall \vec{x} (\varphi(\vec{x}) \leftrightarrow \psi(\vec{x}))$$

The MRDP Theorem means we can eliminate bounded quantifiers from Σ_1-formulas. The following exercise gives its relevance to Hilbert's 10th Problem.

Exercise 97 Show that for every quantifier-free $\mathcal{L}_{\mathrm{PA}}$-formula $\varphi(y,\vec{x})$ there are polynomials $P(y,\vec{x})$ and $Q(y,\vec{x})$ such that for all tuples $\vec{n}$ of natural numbers: $\mathcal{N} \models \exists y\varphi[y,\vec{n}]$ if and only if the equation $P(y,\vec{n}) = Q(y,\vec{n})$ has a solution in $\mathbb{N}$.

Corollary 6.12 *Any inclusion between models of* PA *is Δ_0-elementary.*

Proof. Let $\theta(\vec{x})$ be Δ_0. Since both θ and $\neg\theta$ are Σ_1, by the MRDP Theorem there are quantifier-free formulas φ and ψ such that

$$\begin{array}{l} \mathrm{PA} \vdash \forall\vec{x}(\theta(\vec{x}) \leftrightarrow \exists\vec{y}\varphi(\vec{x},\vec{y})) \\ \mathrm{PA} \vdash \forall\vec{x}(\neg\theta(\vec{x}) \leftrightarrow \exists\vec{z}\psi(\vec{x},\vec{z})) \end{array}$$

Now let $\mathcal{M}_1 \subseteq \mathcal{M}_2$ be an inclusion of models of PA. If, for $\vec{a} \in \mathcal{M}_1$, $\mathcal{M}_1 \models \theta[\vec{a}]$ then for certain $\vec{b} \in \mathcal{M}_1$, $\mathcal{M}_1 \models \varphi[\vec{a},\vec{b}]$. Since φ is quantifier-free, $\mathcal{M}_2 \models \varphi[\vec{a},\vec{b}]$ and so $\mathcal{M}_2 \models \theta[\vec{a}]$, since $\mathcal{M}_2$ is a model of PA. The argument in the other direction uses the equivalence for $\neg\theta$, and is the same. ■

Theorem 6.13 (Gaifman's Splitting Theorem) *Let $\mathcal{M}_1 \subseteq \mathcal{M}_2$ be an inclusion of models of* PA. *Then there is a unique model K with $\mathcal{M}_1 \subseteq_{\mathrm{cf}} K \subseteq_e \mathcal{M}_2$. Moreover, $\mathcal{M}_1 \prec K$, so K is a model of* PA *too.*

Proof. Clearly, there is at most one K with $\mathcal{M}_1 \subseteq_{\mathrm{cf}} K \subseteq_e \mathcal{M}_2$; we have to take

$$K = \{m \in \mathcal{M}_2 \,|\, \text{for some } n \in \mathcal{M}_1, m < n\}$$

Then K is a $\mathcal{L}_{\mathrm{PA}}$-substructure of $\mathcal{M}_2$, as well as an initial segment of it, so K is a model of PA^- and $K \prec_{\Delta_0} \mathcal{M}_2$. Since $\mathcal{M}_1 \prec_{\Delta_0} \mathcal{M}_2$ by Corollary 6.12, also $\mathcal{M}_1 \prec_{\Delta_0} K$ (check this!). Theorem 6.10 now gives $\mathcal{M}_1 \prec K$. ■

Corollary 6.14 *Every nonstandard model of* PA *has proper elementary cofinal extensions.*

Proof. Let $\mathcal{M}$ be a nonstandard model of PA. Let $\mathcal{L}'$ be $\mathcal{L}_{\mathrm{PA}}$ augmented with constants $\underline{m}$ for every $m \in \mathcal{M}$, as well as a new constant c. Let $b \in \mathcal{M}$ be nonstandard and consider the theory

$$\mathrm{Th}(\mathcal{M}) \cup \{c \neq \underline{m} \,|\, m \in \mathcal{M}\} \cup \{c < \underline{b}\}$$

By compactness, this theory has a model $\mathcal{M}'$ which is an elementary extension of $\mathcal{M}$; applying theorem 6.13 to the inclusion $\mathcal{M} \subseteq \mathcal{M}'$ gives $\mathcal{M} \subseteq_{\mathrm{cf}} K \subseteq_e \mathcal{M}'$ with $\mathcal{M} \prec K$. Moreover, $c \in K \setminus \mathcal{M}$, so the extension is proper. ■

6.5 Prime Models and Elementary End-extensions

In this section we shall ultimately see that every model $\mathcal{M}$ of PA has a proper elementary end-extension. For *countable* $\mathcal{M}$, this is a relatively easy Omitting Types argument, given below; but the general case needs a more sophisticated approach. We shall review the theory of prime models of complete theories extending PA, and then, by a rather tricky argument, find a proper elementary end-extension of any given model $\mathcal{M}$ as a particular prime model. First, let us do the countable case. From now on, $\mathcal{L}_{\mathrm{PA}}(\mathcal{M})$ always denotes the language $\mathcal{L}_{\mathrm{PA}}$ augmented with constants from the model $\mathcal{M}$. Let c be a new constant, and consider, in the language $\mathcal{L}_{\mathrm{PA}}(\mathcal{M}) \cup \{c\}$, the theory $T_{\mathcal{M}}(c)$:

$$T_{\mathcal{M}}(c) = \{\theta \in \mathcal{L}_{\mathrm{PA}}(\mathcal{M}) \mid \mathcal{M} \models \theta\} \cup \{c > m \mid m \in \mathcal{M}\}$$

For every $a \in \mathcal{M}$, let $\Sigma_a(x)$ be the type

$$\Sigma_a(x) = \{x < a\} \cup \{x \neq b \mid b \in \mathcal{M}\}$$

Every model of $T_{\mathcal{M}}(c)$ is a proper elementary extension of $\mathcal{M}$, and it is an end-extension if and only if it omits each $\Sigma_a(x)$. Since $\mathcal{M}$ is countable, we may, by the Extended Omitting Types Theorem, conclude that there is such a model, provided we can show that $T_{\mathcal{M}}(c)$ locally omits each $\Sigma_a(x)$.

Suppose that there is an $\mathcal{L}_{\mathrm{PA}}(\mathcal{M})$-formula $\varphi(u, v)$ such that:

$$\begin{array}{ll} (1) & T_{\mathcal{M}}(c) \vdash \varphi(u,c) \to u < a \\ (2) & \text{For all } b \in \mathcal{M} : T_{\mathcal{M}}(c) \vdash \varphi(u,c) \to u \neq b \end{array}$$

By definition of $T_{\mathcal{M}}(c)$, (1) implies that there is $n_1 \in \mathcal{M}$ such that

$$(3) \quad \mathcal{M} \models \forall x > n_1 \forall u(\varphi(u,x) \to u < a))$$

And similarly (2) implies that for every $b \in \mathcal{M}$ there is $n_b \in \mathcal{M}$ such that $\mathcal{M} \models \forall x > n_b \forall u(\varphi(u,x) \to u \neq b))$. By induction in $\mathcal{M}$, it follows that

$$(4) \quad \mathcal{M} \models \forall z \exists y \forall x > y \forall u(\varphi(u,x) \to u > z))$$

If n_2 is such that $\mathcal{M} \models \forall x > n_2 \forall u(\varphi(u,x) \to u > a))$, then for $n = \max(n_1, n_2)$ we have

$$\mathcal{M} \models \forall x > n \forall u \neg\varphi(u,x)$$

and therefore, $T_{\mathcal{M}}(c) \vdash \forall u \neg\varphi(u,c)$. So we see that our assumption leads to the conclusion that $\varphi(u,c)$ is inconsistent with $T_{\mathcal{M}}(c)$, which therefore locally omits $\Sigma_a(x)$.

Since the Omitting Types theorem is false for uncountable languages and for uncountably many types (see, e.g., Chang & Keisler), the general case turns out to be more complicated.

6.5.1 Prime Models

Let $\mathcal{M}$ be a model of PA and $A \subseteq \mathcal{M}$. By $K(\mathcal{M}; A)$ we denote the set of elements of $\mathcal{M}$ which are definable over A. That is, those elements a for which there is a formula $\theta_a(x, u_1, \ldots, u_n)$ of $\mathcal{L}_{\mathrm{PA}}$ and elements $a_1, \ldots, a_n \in A$ such that

$$\mathcal{M} \models \forall x(\theta_a(x, a_1, \ldots, a_n) \leftrightarrow x = a)$$

Let $\mathcal{L}_{\mathrm{PA}}(A)$ the language with constants from A added, and $\mathrm{Th}(\mathcal{M})_A$ the $\mathcal{L}_{\mathrm{PA}}(A)$-theory which is true in $\mathcal{M}$.

Theorem 6.15

i) $K(\mathcal{M}; A)$ *is an* $\mathcal{L}_{\mathrm{PA}}(A)$*-substructure of* $\mathcal{M}$*, and* $A \subseteq \mathcal{L}_{\mathrm{PA}}(A) \prec \mathcal{M}$ *as* $\mathcal{L}_{\mathrm{PA}}(A)$*-structures;*

ii) *For every model* $\mathcal{M}'$ *of* $\mathrm{Th}(\mathcal{M})_A$ *there is a unique* $\mathcal{L}_{\mathrm{PA}}(A)$*-elementary embedding from* $K(\mathcal{M}; A)$ *into* $\mathcal{M}'$*;*

iii) $K(\mathcal{M}; A)$ *has no proper* $\mathcal{L}_{\mathrm{PA}}(A)$*-elementary substructures and no non-trivial* $\mathcal{L}_{\mathrm{PA}}(A)$*-automorphisms.*

Proof. i) Certainly $A \subseteq K(\mathcal{M}; A)$ since every $a \in A$ is defined over A by the formula $x = a$. If a and b are defined by $\mathcal{L}_{\mathrm{PA}}(A)$-formulas $\theta_a(x)$ and $\theta_b(x)$ respectively, then $a + b$ is defined by $\exists zw(\theta_a(z) \wedge \theta_b(w) \wedge x = z + w)$; similarly $a{\cdot}b$ is defined over A. So $K(\mathcal{M}; A)$ is an $\mathcal{L}_{\mathrm{PA}}(A)$-substructure of $\mathcal{M}$. To see that $K(\mathcal{M}; A) \prec \mathcal{M}$ we employ the Tarski-Vaught test. Let $\exists x\varphi$ be an $\mathcal{L}_{\mathrm{PA}}(A)$-sentence which is true in $\mathcal{M}$. Since $\mathcal{M}$ satisfies the least number principle, we have

$$\mathcal{M} \models \exists x(\varphi(x) \wedge \forall y < x \neg\varphi(y))$$

The formula $\varphi(x) \wedge \forall y < x \neg\varphi(y)$ now defines an element of $K(\mathcal{M}; A)$ which satisfies φ, so $K(\mathcal{M}; A) \models \exists x\varphi$

ii) For every $a \in K(\mathcal{M}; A)$ let $\theta_a(x)$ be an $\mathcal{L}_{\mathrm{PA}}(A)$-formula defining a. For a model $\mathcal{M}'$ of $\mathrm{Th}(\mathcal{M})_A$, send a to the unique element a' of $\mathcal{M}'$ such that $\mathcal{M}' \models \theta_a(a')$. This defines a mapping $h : K(\mathcal{M}; A) \to \mathcal{M}'$. This does not depend on the choices for θ_a, because if a is also defined by ζ_a, then $\mathcal{M}$ and $\mathcal{M}'$ satisfy the formula $\forall x(\theta_a(x) \leftrightarrow \zeta_a(x))$. One sees that h is an embedding of $\mathcal{L}_{\mathrm{PA}}(A)$-structures, and the proof that it is elementary, is by a similar application of the Tarski-Vaught test as in i). Finally, h must be unique with these properties, since $h(a)$ must satisfy $\theta_a(x)$.

iii) Since every $\mathcal{L}_{\mathrm{PA}}(A)$-automorphism of $K(\mathcal{M};A)$ is an $\mathcal{L}_{\mathrm{PA}}(A)$-elementary embedding, there can be at most one such by ii); so the identity function is the only one.

If $\mathcal{M}' \prec K(\mathcal{M};A)$ is a proper $\mathcal{L}_{\mathrm{PA}}(A)$-elementary substructure, by ii) there is a unique $\mathcal{L}_{\mathrm{PA}}(A)$-elementary embedding $h : K(\mathcal{M};A) \to \mathcal{M}'$. Composing with the identity gives an elementary embedding of $K(\mathcal{M};A)$ into itself. By ii), there is only one such, which is the identity. But this cannot factor through a proper subset, of course. ■

From the proof of theorem 6.15 we see that if $\mathcal{M}'$ is a model of $\mathrm{Th}(\mathcal{M})_A$ and $A' \subseteq \mathcal{M}'$ is the set of interpretations of the constants from A, then the unique $h : K(\mathcal{M};A) \to \mathcal{M}'$ takes values in $K(\mathcal{M}';A')$. By symmetry, we must have that the models $K(\mathcal{M};A)$ and $K(\mathcal{M}';A')$ are isomorphic. Therefore, the model $K(\mathcal{M};A)$ is determined by the theory $\mathrm{Th}(\mathcal{M})_A$, and does not depend on $\mathcal{M}$ or A.

If $A = \emptyset$, we write $K(\mathcal{M})$ for $K(\mathcal{M};A)$. In view of the remark above, for every consistent, complete $\mathcal{L}_{\mathrm{PA}}$-theory T extending PA we have a prime model K_T which we can take to be $K(\mathcal{M})$ for any model $\mathcal{M}$ of T.

Exercise 98 This exercise recalls some notions from Model Theory. Given a complete theory T in a countable language $\mathcal{L}$, we say that an $\mathcal{L}$-formula $\varphi(x_1,\ldots,x_n)$ is *complete* in T if it is consistent with T and for any other $\mathcal{L}$-formula $\psi(x_1,\ldots,x_n)$, either $T \vdash \forall x_1 \cdots x_n(\varphi(\vec{x}) \to \psi(\vec{x}))$ or $T \vdash \forall x_1 \cdots x_n(\varphi(\vec{x}) \to \neg\psi(\vec{x}))$ (Equivalently, $T \cup \{\varphi(c_1,\ldots,c_n)\}$ is a complete $\mathcal{L}\cup\{c_1,\ldots,c_n\}$-theory, where $c_1,\ldots,c_n$ are new constants). The theory T is called *atomic* if for every $\mathcal{L}$-formula $\varphi(\vec{x})$ which is consistent with T, there is a complete formula $\psi(\vec{x})$ such that $T \vdash \forall\vec{x}(\psi(\vec{x}) \to \varphi(\vec{x}))$.

Show that every complete extension of PA is atomic.

6.5.2 Conservative Extensions and MacDowell-Specker Theorem

The MacDowell-Specker Theorem asserts what we announced as our main result for this section: every model of PA has a proper elementary end-extension. The way we shall prove it, it comes out as a corollary of another theorem.

If $\mathcal{M}_1 \subseteq \mathcal{M}_2$ is an inclusion of models of PA, we say that $\mathcal{M}_2$ is a *conservative extension* of $\mathcal{M}_1$, if for every subset X of $\mathcal{M}_2$, if X is definable in $\mathcal{M}_2$ in parameters from $\mathcal{M}_2$ (that is: there is $\theta(x,u_1,\ldots,u_n)$ and $a_1,\ldots,a_2 \in \mathcal{M}_2$ such that $X = \{m \in \mathcal{M}_2 \,|\, \mathcal{M}_2 \models \theta(x,a_1,\ldots,a_n)\}$) then $X \cap \mathcal{M}_1$ is definable in $\mathcal{M}_1$ in parameters from $\mathcal{M}_1$.

The theorem we shall prove, is:

Theorem 6.16 *Every model of* PA *has a proper elementary conservative extension.*

Let us see that this implies what we want:

Lemma 6.17 *Every conservative extension is an end-extension.*

Proof. Let $\mathcal{M}_1 \subseteq \mathcal{M}_2$ a conservative extension; let $a \in \mathcal{M}_1, b \in \mathcal{M}_2$ and suppose $b < a$. The set $\{m \in \mathcal{M}_2 \,|\, m \leq b\}$ is clearly definable in $\mathcal{M}_2$ with parameter b, so $\{m \in \mathcal{M}_1 \,|\, m \leq b\}$ is definable in parameters from $\mathcal{M}_1$, say

$$\{m \in \mathcal{M}_1 \,|\, m \leq b\} = \{m \in \mathcal{M}_1 \,|\, \mathcal{M}_1 \models \theta(m, a_1, \ldots, a_n)\}$$

Since $a \in \mathcal{M}_1$ and $b < a$ we have $\mathcal{M}_1 \models \forall x(\theta(x, a_1, \ldots, a_n) \to x \leq a)$. By the least number principle in $\mathcal{M}_1$, there is a least $a' \in \mathcal{M}_1$ such that

$$\mathcal{M}_1 \models \forall x(\theta(x, a_1, \ldots, a_n) \to x \leq a')$$

It follows that $\mathcal{M}_1 \models \theta(a', a_1, \ldots, a_n)$, so $a' \leq b$. But if $a' < b$ then $a'+1 \leq b$ whence $\mathcal{M}_1 \models \theta(a'+1, a_1, \ldots, a_n)$, but of course $\mathcal{M}_1 \not\models a'+1 \leq a'$. Therefore we must have $a' = b$, so $b \in \mathcal{M}_1$, as desired. ∎

Hence, for the record:

Corollary 6.18 (MacDowell-Specker) *Every model of* PA *has a proper elementary end-extension.*

We now embark on the proof of theorem 6.16. We introduce the abbreviation $Qx\,\varphi(x)$ for $\forall y \exists x(x > y \wedge \varphi(x))$ ("there exist unboundedly many x satisfying $\varphi(x)$").

Lemma 6.19 *Let $\mathcal{M}$ be a model of* PA*, $\varphi(x)$ an $\mathcal{L}_{\mathrm{PA}}(\mathcal{M})$-formula such that $\mathcal{M} \models Qx\varphi(x)$, and $\theta(x, y)$ an arbitrary $\mathcal{L}_{\mathrm{PA}}(\mathcal{M})$-formula. Then there is an $\mathcal{L}_{\mathrm{PA}}(\mathcal{M})$-formula $\psi(x)$ with the properties:*

i) $\mathcal{M} \models Qx\psi(x)$

ii) $\mathcal{M} \models \forall x(\psi(x) \to \varphi(x))$

iii) $\mathcal{M} \models \forall y \neg(Qx(\psi(x) \wedge \theta(x, y)) \wedge Qx(\psi(x) \wedge \neg\theta(x, y)))$

Proof. An equivalent for item iii) is:

$$\mathcal{M} \models \forall y \exists z (\forall x > z(\psi(x) \to \theta(x,y)) \vee \forall x > z(\psi(x) \to \neg\theta(x,y)))$$

The idea of the proof is as follows. We shall construct an $\mathcal{L}_{\mathrm{PA}}(\mathcal{M})$-formula $\chi(y,x)$ such that

$$\begin{array}{ll} (1) & \mathcal{M} \models \forall x(\chi(0,x) \leftrightarrow (\varphi(x) \wedge (\theta(x,0) \leftrightarrow Qv(\varphi(v) \wedge \theta(v,0))))) \\ (2) & \mathcal{M} \models \forall yx(\chi(y+1,x) \leftrightarrow \chi(y,x) \wedge (\theta(x,y+1) \leftrightarrow Qv(\chi(y,v) \wedge \theta(v,y+1)))) \end{array}$$

For the moment, assume that $\chi(y,x)$ has been defined. It follows, by induction in $\mathcal{M}$, that $\mathcal{M} \models \forall y Qx\chi(y,x)$; for $Qx\chi(y,x)$ implies $Qx(\chi(y,x) \wedge \theta(x,y+1)) \vee Qx(\chi(y,x) \wedge \neg\theta(x,y+1))$, so $Qx\chi(y+1,x)$. We note also, that $\mathcal{M} \models \forall yx(\chi(y,x) \to \varphi(x) \wedge \forall v \leq y\chi(v,x))$.

In order to define $\psi(x)$ from $\chi(y,x)$ we use theorem 4.9. We write $(s)_i$ instead of $(a,m)_i$ as in that theorem, putting $s = j(a,m)$:

$$(s)_i = \mathrm{rm}(j_1(s), (i+1)j_2(s)+1)$$

Let us also write $x = \mu z\varphi(z)$ for $\varphi(x) \wedge \forall y < x\neg\varphi(y)$.

Since $\forall y Qx\chi(y,x)$ holds in $\mathcal{M}$, we have by induction on z and theorem 4.9 that the sentence

$$\forall z \exists s((s)_0 = \mu x\chi(0,x) \wedge \forall i < z((s)_{i+1} = \mu x(x > (s)_i \wedge \chi(i+1,x))))$$

is true in $\mathcal{M}$; write this as $\forall z \exists s \Phi(z,s)$. Define

$$(3) \qquad \psi(x) \equiv \exists s(\Phi(x,s) \wedge \exists i \leq x(s)_i = x)$$

Then $\mathcal{M} \models Qx\psi(x)$, so statement i) of the Lemma is satisfied. Statement ii), that $\forall x(\psi(x) \to \varphi(x))$, follows from $\forall yx(\chi(y,x) \to \varphi(x))$. As to statement iii), first note that if $w \leq z \wedge \Phi(z,s) \wedge \Phi(w,t)$, then $\forall v \leq w((s)_v = (t)_v)$. So for all $z \geq y$, if $\Phi(z,s)$ then $\forall w(y \leq w \leq z \to \chi(y,(s)_w)$. So if $\Phi(y,s) \wedge \psi(x) \wedge x \geq (s)_y$ then $\theta(y,x) \leftrightarrow \theta(y,(s)_y)$, which ensures that statement iii) holds.

It remains to define the formula $\chi(y,x)$ and prove the equivalences (1) and (2). Again, we use the sequence coding $(s)_i$. Let $P(s,y)$ be the formula

$$\forall u \leq y((s)_u = 0 \leftrightarrow Qz(\varphi(z) \wedge \theta(z,u) \wedge \forall v < u(\theta(z,v) \leftrightarrow (s)_v = 0)))$$

and define $\chi(y,x)$ as

$$\exists s(P(s,y) \wedge \forall u \leq y(\theta(x,u) \leftrightarrow (s)_u = 0) \wedge \varphi(x))$$

Since $P(s,0) \leftrightarrow ((s)_0 = 0 \leftrightarrow Qz(\varphi(z) \wedge \theta(z,0)))$, we have

$$\psi(0,x) \leftrightarrow \varphi(x) \wedge (\theta(x,0) \leftrightarrow Qz(\varphi(z) \wedge \theta(z,0)))$$

so (1) holds.

For (2), first note that $P(s,y) \wedge P(t,y)$ implies $\forall u \leq y((s)_u = 0 \leftrightarrow (t)_u = 0)$; from this and the definition of $\chi(y,x)$ it follows directly that

$$\begin{aligned} P(s,y) \rightarrow \forall u \leq y \forall x(\psi(u,x) \leftrightarrow \\ \varphi(x) \wedge \forall v \leq u(\theta(x,v) \leftrightarrow (s)_v = 0)) \end{aligned} \tag{4}$$

holds. We prove the equivalence of (2):

$\rightarrow$: Suppose $\chi(y+1,x)$, so

$$P(s,y+1) \wedge \forall u \leq y+1(\theta(x,u) \leftrightarrow (s)_u = 0) \wedge \varphi(x)$$

for some s. Applying (4) with $y+1$ for y we have

$$\forall z(\chi(y+1,z) \leftrightarrow \varphi(z) \wedge \forall v \leq y+1(\theta(z,v) \leftrightarrow (s)_v = 0))$$

so $\varphi(x) \wedge (\theta(x,y+1) \leftrightarrow (s)_{y+1} = 0)$. Combining this with the definition of $P(s,y+1)$, the fact that $\chi(y,x)$ implies $\varphi(x) \wedge \forall v \leq y \chi(v,x)$, and applying (4) again (inside the part $Qz(\ldots)$), we get

$$\chi(y,x) \wedge (\theta(x,y+1) \leftrightarrow Qz(\theta(z,y+1) \wedge \chi(y,z)) \tag{5}$$

$\leftarrow$: Conversely, assume (5) and $P(s,y)$. By theorem 4.9 there is t such that $\forall u \leq y((s)_u = (t)_u$, and

$$(t)_{y+1} = 0 \leftrightarrow Qz(\varphi(z) \wedge \theta(z,y+1) \wedge \forall v \leq y(\theta(z,v) \leftrightarrow (s)_v = 0))$$

Then $P(t,y+1)$ holds. We have to show:

$$\forall u \leq y+1(\theta(x,u) \leftrightarrow (t)_u = 0) \wedge \varphi(x)$$

Since $\chi(y,x)$ we have $\varphi(x)$, and for $u \leq y$ this is clear, since $P(s,y)$. For $u = y+1$ we have:

$$\begin{aligned} \theta(x,y+1) &\leftrightarrow Qz(\theta(z,y+1) \wedge \chi(y,z)) \\ &\leftrightarrow Qz(\varphi(z) \wedge \theta(z,y+1) \wedge \forall v \leq y \\ &\qquad (\theta(z,v) \leftrightarrow (t)_v = 0)) \\ &\leftrightarrow (t)_{y+1} = 0 \end{aligned}$$

(the first equivalence by (5); the second by (4); the third by definition of t) We have proved the equivalence (2), and hence the lemma. ∎

We finish the proof of Theorem 6.16. Fix an enumeration $\theta_0(c, \vec{y}^{(0)}), \theta_1(c, \vec{y}^{(1)}), \ldots$ of all formulas of $\mathcal{L}_{\mathrm{PA}} \cup \{c\}$ (so $\theta_i(x, \vec{y}^{(i)})$ is an $\mathcal{L}_{\mathrm{PA}}$-formula and $\vec{y}^{(i)}$ is the list of free variables of $\theta_i(c, \vec{y}^{(i)})$). We construct a sequence of $\mathcal{L}_{\mathrm{PA}}$-formulas $\varphi_0(x), \varphi_1(x), \ldots$ in one free variable x, such that $\mathcal{M} \models Qx\varphi_i(x)$ for all i, as follows. Put $\varphi_0(x) \equiv x = x$. Given $\varphi_i(x)$ such that $\mathcal{M} \models Qx\varphi_i(x)$, we apply lemma 6.19 to find $\varphi_{i+1}(x)$ such that:

$$\begin{aligned}
&\mathcal{M} \models Qx\,\varphi_{i+1}(x)\\
&\mathcal{M} \models \forall x(\varphi_{i+1}(x) \to \varphi_i(x))\\
&\mathcal{M} \models \forall \vec{y}^{(i)} \exists z(\forall x > z(\varphi_{i+1}(x) \to \theta_i(x, \vec{y}^{(i)})) \vee\\
&\qquad\qquad \forall x > z(\varphi_{i+1}(x) \to \neg\theta_i(x, \vec{y}^{(i)})))
\end{aligned}$$

Consider the $\mathcal{L}_{\mathrm{PA}}(\mathcal{M}) \cup \{c\}$-theory T given by the axioms

$$\begin{aligned}
&\{\theta(\vec{a}) \in \mathcal{L}_{\mathrm{PA}}(\mathcal{M}) \,|\, \mathcal{M} \models \theta(\vec{a})\} \cup\\
&\quad \{c > a \,|\, a \in \mathcal{M}\} \cup \{\varphi_i(c) \,|\, i \in \mathbb{N}\}
\end{aligned}$$

Since every finite subset of this has an interpretation in $\mathcal{M}$, T is consistent. Let $\mathcal{M}'$ be a model of T and let $K = K(\mathcal{M}'; \mathcal{M} \cup \{c\})$. We have $\mathcal{M} \prec \mathcal{M}'$ as $\mathcal{L}_{\mathrm{PA}}(\mathcal{M})$-structures, $\mathcal{M} \subseteq K$ and $K \prec \mathcal{M}'$ as $\mathcal{L}_{\mathrm{PA}}(\mathcal{M}) \cup \{c\}$-structures; it follows that $\mathcal{M} \prec K$ as $\mathcal{L}_{\mathrm{PA}}(\mathcal{M})$-structures. Also, $c \in K \setminus \mathcal{M}$, so K is a proper elementary extension of $\mathcal{M}$. We want to show that the extension $\mathcal{M} \subseteq K$ is conservative.

Suppose s subset $S \subseteq K$ is defined by $S = \{k \,|\, K \models \theta(k, b_1, \ldots b_n)\}$ with $b_1, \ldots, b_n \in K$. By definition of K, every b_i is defined in $\mathcal{M}'$ by a formula $\eta_i(v, a_1, \ldots, a_k, c)$ with $a_1, \ldots, a_k \in \mathcal{M}$. Now the formula

$$\exists v_1 \cdots v_n (\bigwedge_{i=1}^{n} \eta_i(v_i, y_1, \ldots, y_k, x) \wedge \theta(y_0, v_1, \ldots, v_n))$$

is an $\mathcal{L}_{\mathrm{PA}}$-formula, so occurs in our enumeration as $\theta_j(x, \vec{y}^{(j)})$, with $\vec{y}^{(j)} = y_0, \ldots, y_k$. We claim:

$$\begin{aligned}
&d \in \mathcal{M} \cap S \Leftrightarrow\\
\mathcal{M} \models\ &\exists w \forall x > w(\varphi_{j+1}(x) \to \theta_j(x, d, a_1, \ldots, a_k)))
\end{aligned}$$

so that $\mathcal{M} \cap S$ is definable in $\mathcal{M}$ over $\mathcal{M}$. Observe, that for $d \in \mathcal{M}$, $d \in S$ if and only if $K \models \theta(d, b_1, \ldots, b_1)$, if and only if $K \models \theta_j(c, d, a_1, \ldots, a_k)$.

By construction of φ_{j+1}, we have either

$$\text{i)}\quad \mathcal{M} \models \exists w \forall x > w(\varphi_{j+1}(x) \rightarrow \theta_j(x, d, a_1, \ldots, a_k))$$

or

$$\text{ii)}\quad \mathcal{M} \models \exists w \forall x > w(\varphi_{j+1}(x) \rightarrow \neg\theta_j(x, d, a_1, \ldots, a_k))$$

These are formulas with parameters in $\mathcal{M}$, so since $\mathcal{M} \prec K$, each one is satisfied in $\mathcal{M}$ if and only if it holds in K. So, i) is the case if and only if $K \models \theta_j(c, d, a_1, \ldots, a_k)$, if and only if $d \in S$, as desired.

Chapter 7

Further reading

Logic is a very old topic, but one can argue that its development as a mathematical discipline starts with Gödel's Completeness Theorem (theorem 2.6), published in 1930 ([7]). However, most textbooks in Logic use the proof of the same result, that was later given by Leon Henkin ([11]).

One year after the Completeness Theorem, Gödel published the First Incompleteness Theorem in [8]. An English translation of this paper is in the booklet [9] and also in [5], which is a very nice and affordable collection of basic, seminal papers by Gödel, Church, Turing, Kleene and Post.

The Second Incompleteness Theorem was announced by Gödel, but first proved by Hilbert and Bernays in [14]. There are many modern expositions of the Incompleteness Theorems: we mention [2, 21, 23]. A classic book with lots of information on arithmetization techniques and subsystems of PA, is [10].

A good read about Gödel's life is the biography [16].

The theory of recursive functions, pioneered by Hilbert and Ackermann, was fully developed in the 1930's by Alonzo Church, Stephen Cole Kleene and Alan Turing. Classic textbooks on this theory are [20, 18, 19]; a more accessible student text is [4]. In the collection [12] one finds papers by distinguished computability theorists on the genesis of the concepts of recursive function theory.

Turing's very interesting life is described in [15].

A very original and well-written book on the number theory that plays a role in this area of Logic, is [22].

For models of PA, we recommend the text book [17].

Finally, Gödel's theorems have occupied many great minds and triggered philosophical debate from different angles. Among the many people who have tried to interpret the theorems from a philosophical or artistic point of view, are Douglas Hofstadter, Morris Kline and Roger Penrose. An extremely well-written, and at places amusing, book about the sense and nonsense of this, is [6]. Warmly recommended if you wish to extend your understanding of the significance of the Icompleteness theorems beyond the technical side.

Bibliography

[1] W. Ackermann. Zum Hilbertschen Aufbau der reellen Zahlen. *Mathematische Annalen*, 99:118–133, 1928.

[2] G. Boolos, J.P. Burgess, and R.C. Jeffrey. *Computability and Logic*. Cambridge University Press, 2007. Fifth edition.

[3] Alonzo Church. A note on the Entscheidungsproblem. *Journal of Symbolic Logic*, 1:40–41, 1936. Correction in same volume, pp. 101–102.

[4] N.J. Cutland. *Computability*. Cambridge University Press, 1980.

[5] Martin Davis (ed). *The Undecidable - Basic Papers on Undecidable Propositions, Unsolvable Problems and Computable Functions*. Dover, 2004. Reprint of 1965 edition by Raven Press Books.

[6] T. Franzén. *Gödel's Theorem - An Incomplete Guide to Its Use and Abuse*. AK Peters, 2005.

[7] K. Gödel. Die Vollständigkeit der Axiome des logischen Funktionenkalküls. *Monatsh.Math.Phys.*, 37:349–360, 1930.

[8] K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 38:173–198, 1931.

[9] K. Gödel. *On Formally Undecidable Propositions of Principia Mathematica and Related Systems*. Dover, 1992. Reprint of 1962 edition by Basic Books; translation of [8].

[10] P Hájek and P. Pudlák. *Metamathematics of First-Order Arithmetic*. Perspectives in Mathematical Logic. Springer, 1993. Second printing 1998.

[11] L. Henkin. The completeness of the first-order functional calculus. *Journal of Symbolic Logic*, 14:159–166, 1949.

[12] R. Herken (ed). *The Universal Turing Machine - A Half-Century Survey*. Oxford University Press, 1988. Collection of papers by Hodges, Kleene, Gandy, Feferman, Davis and others.

[13] D. Hilbert and W. Ackermann. *Grundzügen der theoretischen Logik*. Springer Verlag, 1928.

[14] D. Hilbert and P. Bernays. *Grundlagen der Mathematik I*. Springer Verlag, 1934.

[15] A. Hodges. *Alan Turing: the enigma*. Random House, London, 1992.

[16] John W. Dawson jr. *Logical Dilemmas - The Life and Work of Kurt Gödel*. AK Peters, 1997.

[17] R. Kaye. *Models of Peano Arithmetic*, volume 15 of *Oxford Logic Guides*. Oxford University Press, Oxford, 1991.

[18] P. Odifreddi. *Classical Recursion Theory*, volume 125 of *Studies in Logic*. North-Holland, 1989.

[19] P. Odifreddi. *Classical Recursion Theory II*, volume 143 of *Studies in Logic*. North-Holland, 1999.

[20] H. Rogers. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill, 1967. (reprinted by MIT Press, Cambridge MA, 1987).

[21] P. Smith. *An Introduction to Gödel's Theorems*. Cambridge University Press, 2009. Fourth printing with corrections.

[22] Craig Smoryński. *Logical Number Theory I - An Introduction*. Springer-Verlag, 1991.

[23] R. Smullyan. *Gödel's Incompleteness Theorems*, volume 19 of *Oxford Logic Guides*. Oxford University Press, 1992.

[24] G. Sudan. Sur le nombre transfini ω^ω. *Bulletin Mathématique de la Société Roumaine des Sciences*, 30:11–30, 1927.

Index