How to prove the First and Second Incompleteness Theorem using concatenation

A. Visser, O. de Moor and M.J. Walsteijn

RUU-CS-86-15
September 1986

Rijksuniversiteit Utrecht

Vakgroep informatica

Budapestlaan 6 3584 CD Utrecht
Corr. adres: Postbus 80.012 3508 TA Utrecht
Telefoon 030-53 1454
The Netherlands

How to prove the First and Second Incompleteness Theorem using concatenation

A. Visser, O. de Moor and M.J. Walsteijn

Technical Report RUU-CS-86-15
September 1986

Department of Computer Science
University of Utrecht
P.O. Box 80.012, 3508 TA Utrecht
the Netherlands

PREFACE.

These lecture notes grew during courses in Metamathematics in 1985 and 1986. Their principle aim is to prove Gödel's First and Second Incompleteness Theorem for the system of Peano Arithmetic providing details of the arithmetization to such a degree that they are sufficient for the student to enable him to see what still needs to be done (in principle) and how it should be done.

The main new twist of these notes is the fact that the use of the Chinese Remainder Theorem is avoided. Sequences are built using concatenation. The idea to do this is taken from [Boolos & Jeffrey], who attribute the idea to Kripke. Let me briefly comment on the differences of the treatment here and that of Boolos and Jeffrey:

(i) A minor point is that, due to an infelicity in the definition, the concatenation operator of Boolos and Jeffrey fails to be associative. Let a prime p be given. They define:

$\eta(b) :=$ the smallest power of p that is both greater than b and 1

$a * b := a \cdot \eta(b) + b$

Clearly $(1 * 0) * 1 = p^2 + 1$, but $1 * (0 * 1) = p + 1$. The example also exhibits the sole reason of the failure: 0 doesn't function as the empty word, as it should, when used as the right hand side in a concatenation.

The defect is easily repaired by dropping the "and 1" at the end of the definition of $\eta(b)$. A consequence of the failure of transitivity in Boolos and Jeffrey's treatment is that their definition of "part" doesn't generally the job they want it to do.

(ii) The main point is this: Boolos and Jeffrey restrict their attention to the First Incompleteness Theorem. Because of this they do not need to prove a Prolongation Lemma (see (61) and (62) on pp. 16,17 of this report) in Peano Arithmetic for sequences as they define them. They can build the sequences outside the system. It turns out that their definition of sequence cannot easily be used to prove a Prolongation Lemma: To prolong a sequence a la Boolos and Jeffrey it may happen that one has to change the prime number on which the concatenation is based. To do this one needs primitive recursion, which is not yet justified!

Clearly the present approach has the advantage that one is able at an early stage to embed concatenation into Peano Arithmetic and prove the necessary properties of the representing arithmetical operation. However, surely one would hope that the approach via concatenation would also be simpler than the one employing the β-function. As far as I can see this last aim has not been quite achieved, due to the considerable difficulty to construct sequences from strings.

The present notes need improvement and extension. In subsequent classes I hope to add at least a description of the formalization of the proof of Σ-completeness and a treatment of the theorems of Löb, Rosser and Church.

I thank Hans Mulder for helpful discussions in an early stage of the genesis of these notes. Oege de Moor and Michiel Walsteijn did a wonderful job in converting the rough class notes into readable text.

Albert Visser.
University of Utrecht
Centrale Interfaculteit
Filosofisch Instituut
Heidelberglaan 2
Postbus 80.103
3508 TC Utrecht

1. Peano Arithmetic

1.1. Introduction

To prove the Gödel Theorems, we shall construct a self-referential statement in number theory. Such a statement is like the Epiminides paradox:

This statement is false.

No matter how one reasons, it is not possible to determine the falsity or truth of this sentence. The paradox-like sentence we seek to express has to be something like:

This sentence cannot be proved in number theory.

Of course, we must use the language of arithmetic. This language should have the ability to ventriloquize: It has to talk about its own sentences. Consider a theory of elementary syntax rich enough to describe the language of arithmetic and to prove its relevant properties. If we can translate the language of this theory into the language of arithmetic in such a way that the translations of syntactical theorems are arithmetically provable, our aim will be reached. Such a translation corresponds most naturally with an embedding of the standard model of our theory of syntax into the structure of the natural numbers.

In this chapter, we start with Peano Arithmetic as the original theory. Whether it meets our requirements remains to be seen. It should adequately express number theory, and the above-mentioned technique should be applicable. In the remaining of this notes, we shall refer to Peano Arithmetic as PA.

1.2. Definitions and conventions

1.2.1. The alphabet of PA

Unary function symbol: S
Binary function symbols: $+, \cdot$
Constant symbol: $\underline{0}$
And the ordinary symbols in a first-order theory.

Because PA is intended to be a theory of something very familiar (natural number arithmetic) it is handy to introduce some notations:

$+(x, y)$ will be written as $x+y$
$\cdot(x, y)$ will be written as $x \cdot y$
$S(x)$ will be written as Sx
whenever convenient.

We should like to refer to other numbers than $\underline{0}$, hence we introduce the abbreviation:

$$\underline{n} := SS \cdots S(\underline{0}) \qquad (n\ S's)$$

Or, more neatly defined:

$$\underline{1} := S(\underline{0}), \quad \underline{n+1} := S(\underline{n})$$

Note that 'n' is a meta-variable ranging over the numbers. The underscore could be interpreted as a function from numbers to terms (i.e. syntactical objects).

1.2.2. Axioms

PA has the following axioms:

(1) $(Sx \neq \underline{0})$

(2) $(Sx = Sy) \rightarrow (x = y)$

(3) $x + \underline{0} = x$

$x + Sy = S(x + y)$

(4) $x \cdot \underline{0} = \underline{0}$

$x \cdot Sy = x \cdot y + x$

(5) $(A\,\underline{0} \wedge \forall x(Ax \rightarrow A\ Sx)) \rightarrow \forall x\ Ax$

This last clause is not really an axiom. It is an axiom schema: for each formula A, an axiom is generated. The schema is called 'The principle of mathematical induction'. One could express it as a natural-deduction rule.

$$\frac{\begin{array}{cc} & [Ax] \\ & D \\ A\underline{0} & ASx \end{array}}{Ax}$$

With the restriction, that x should not occur free in the open hypotheses. Note that $\forall x\ Ax$ can be concluded.

1.3. Basic properties of PA

In paragraph 1.3 we shall prove some basic properties of PA. Considering the amount of properties we are going to prove and the frequent use of proof by induction, we'll introduce a shorthand for it. This shorthand only shows the crux of the proof, leaving out the cosmetic details, which are left to the reader. If the reader wishes a full conventional proof by induction, he should be able to convert the shorthand to the desired shape by himself.

We'll now introduce the shorthand by an example:
Suppose we want to prove that for any

$$n \geq 0,\ 1 + 2 + \cdots + n = \frac{(n^2 + n)}{2}$$

The shorthand looks like this:

- $0 = (0^2 + 0)/2$
- $1 + 2 + \cdots + n + (n+1) =$

  $(1 + 2 + 3 + \cdots + n) + (n+1) = (I.H.)$

  $\frac{n^2 + n}{2} + (n+1) =$

  $\frac{n^2 + n + 2n + 2}{2} =$

  $\frac{n^2 + 2n + 1 + n + 1}{2} =$

  $\frac{(n+1)^2 + (n+1)}{2}$

The conventional proof would be like this:
Basis Step: Let $n = 0$. Then the sum on the left is zero, since there is nothing to add. The expression on the right is $(0^2 + 0)/2$. Hence, the claim holds for $n = 0$.

Induction Step: The induction hypothesis: We assume that for $n = m \geq 0$ the claim holds.
Then

$1 + 2 + \cdots + m + (m+1) =$

$\cdots =$ (By the induction hypothesis)

$\frac{(m+1)^2+(m+1)}{2}$

Therefore the claim holds for $n = m + 1$. By mathematical induction the claim holds for all $n \geq 0$.

Notice that in the shorthand the induction hypothesis is implicit. If we have an induction over more than one variable, we will simply indent a few blanks when the induction over the next variable begins.

We start with an almost trivial, but important property: Any number is either zero or it is the successor of another number.

1) $x = \underline{0} \vee \exists y\ x = Sy$

- $\underline{0} = \underline{0} \vee \exists y\ \underline{0} = Sy$
- $Sx = \underline{0} \vee \exists y\ Sx = Sy$

Properties of addition

In this section, we shall see that addition in PA satisfies our intuitions: It is commutative and associative.

To prove that '+' is commutative, we need the following weaker result:
2) $x+Sy = Sx+y$

- $x + S\underline{0} = S(x + \underline{0})$
  $= Sx$
  $= Sx + \underline{0}$
- $x + SSy = S(x + Sy)$
  $=^{IH} S(Sx + y)$
  $= Sx + Sy$

Commutativity of addition:
3) $x+y = y+x$

- $x + \underline{0} = \underline{0} + x$:
  - $\underline{0} + \underline{0} = \underline{0} + \underline{0}$
  - $Sx + \underline{0} = Sx$
    $= S(x + \underline{0})$
    $=^{IH} S(\underline{0} + x)$
    $= \underline{0} + Sx$
- $x + Sy = Sy + x$:
  $x + Sy = S(x + y)$
  $=^{IH} S(y + x)$
  $= y + Sx$
  $=^{2} Sy + x$

Associativity of addition:
4) $(x+y)+z = x+(y+z)$

- $(x + y) + \underline{0} = x + y$

$$= x + (y + \underline{0})$$

– $(x + y) + Sz = S((x + y) + z)$
$$=^{IH} S(x + (y + z))$$
$$= x + S(y + z)$$
$$= x + (y + Sz)$$

5) $y+\underline{1} = Sy$
An exercise for the reader.

Properties of multiplication

Now we examine the behaviour of $\cdot$. $\underline{1}$ is indeed the multiplicative identity. Distributivity of multiplication over addition holds, and $\cdot$ is both commutative and associative. We employ the following common convention: $x \cdot y$ may be abbreviated to xy.

One is the multiplicative identity:
6) $\underline{1} \cdot x = x$

– $\underline{1} \cdot \underline{0} = \underline{0}$
– $\underline{1} \cdot Sx = \underline{1} \cdot x + \underline{1}$
$$=^{IH} x + \underline{1}$$
$$=^{5} Sx$$

Multiplication is distributive over addition:
7) $(x+y) \cdot z = x \cdot z + y \cdot z$

– $(x + y) \cdot \underline{0} = x \cdot \underline{0} + y \cdot \underline{0}$
– $(x + y) \cdot Sz = (x + y) \cdot z + (x + y)$
$$=^{IH} (x \cdot z + y \cdot z) + (x + y)$$
$$=^{3,4} (x \cdot z + x) + (y \cdot z + y)$$
$$= x \cdot Sz + y \cdot Sz$$

Multiplication is commutative:
8) $x \cdot y = y \cdot x$

– $x \cdot \underline{0} = \underline{0} \cdot x$:
  – $\underline{0} \cdot \underline{0} = \underline{0} \cdot \underline{0}$
  – $Sx \cdot \underline{0} = \underline{0}$
  $$= \underline{0} + \underline{0}$$
  $$=^{IH} \underline{0} \cdot x + \underline{0}$$
  $$= \underline{0} \cdot Sx$$
– $x \cdot Sy = Sy \cdot x$:
$x \cdot Sy = x \cdot y + x$
$$=^{IH} y \cdot x + x$$
$$=^{6} y \cdot x + \underline{1} \cdot x$$
$$=^{7} (y + \underline{1}) x$$
$$=^{5} Sy \cdot x$$

Multiplication is associative:
9) $(x \cdot y) \cdot z = x \cdot (y \cdot z)$

– $(x \cdot y) \cdot \underline{0} = x \cdot (y \cdot \underline{0})$
– $(x \cdot y) \cdot Sz = (x \cdot y) \cdot z + (x \cdot y)$
$$=^{IH} x \cdot (y \cdot z) + x \cdot y$$
$$=^{7,8} x \cdot (y \cdot z + y)$$
$$= x \cdot (y \cdot Sz)$$

Addition, multiplication and equality:

In normal life, there are some special facts concerning multiplication, addition and equality. If the sum of two natural numbers is zero, they are both zero. If the product of two numbers is zero, at least one of them is equal to zero. The only divisor of one is one.

10) $x+y = \underline{0} \rightarrow x = \underline{0} \wedge y = \underline{0}$

Suppose $x + y = \underline{0}$.

$y = Su \rightarrow x + Su = \underline{0}$

$\rightarrow S(x + u) = \underline{0}$

Contradiction. Using result 1, we conclude that $y = \underline{0}$

$x = Sv \rightarrow Sv + y = \underline{0}$

$\rightarrow v + Sy = \underline{0}$

$\rightarrow S(v + y)\underline{0}$

Contradiction. Using result 1, we conclude that $x = \underline{0}$

11) $x+z = y+z \rightarrow x = y$

– $x + \underline{0} = y + \underline{0} \rightarrow x = y$

– $x + Sz = y+Sz \rightarrow S(x + z) = S(y + z)$

$\rightarrow x + z = y + z$

$\rightarrow^{IH} x = y$

12) $x \cdot y = \underline{0} \rightarrow x = \underline{0} \vee y = \underline{0}$

Suppose $x \cdot y = \underline{0}$.

$x = Su \wedge y = Sv \rightarrow Su \cdot Sv = \underline{0}$

$\rightarrow Su \cdot v + Su = \underline{0}$

$\rightarrow S(Su \cdot v + u) = \underline{0}$

Contradiction. Again by 1, we may conclude that $x = \underline{0} \vee y = \underline{0}$

If two numbers, each multiplied by the same non-zero number, yield the same product, they are equal:

13) $x \cdot Sz = y \cdot Sz \rightarrow x = y$

– $\underline{0} \cdot Sz = y \cdot Sz \rightarrow \underline{0} = y \cdot Sz$

$\rightarrow^{8} y = \underline{0} \vee Sz = \underline{0}$

$\rightarrow y = \underline{0}$

– *CASE I* : $y = \underline{0}$, as above.

CASE II : $\neg (y = \underline{0})$, by 1 there is a w such that $y = Sw$. We have:

$Sx \cdot Sz = Sw \cdot Sz \rightarrow^{8} Sz \cdot Sx = Sz \cdot Sw$

$\rightarrow Sz \cdot x + Sz = Sz \cdot w + Sz$

$\rightarrow^{11} Sz \cdot x = Sz \cdot w$

$\rightarrow^{8} x \cdot Sz = w \cdot Sz$

$\rightarrow^{IH} x = w$

$\rightarrow Sx = y$

14) $x \cdot y = \underline{1} \rightarrow x = \underline{1} \wedge y = \underline{1}$

It is immediately clear that $x \neq \underline{0}$, $y \neq \underline{0}$,

Let $x = Su$, $y = Sv$. (By 1)

$Su \cdot Sv = Su \cdot v + Su = S(Su \cdot v + u) = \underline{1}$

Hence $Su \cdot v + u = \underline{0}$.

We get $Su \cdot v = \underline{0} \wedge u = \underline{0}$. By 13, $v = \underline{0}$ too.

Comparison of numbers

Having examined additon and multiplication in some detail, we now turn to the comparison of numbers in PA. First we define two binary predicate symbols, representing the usual order relations in ω.

Define:
$x < y : \Leftrightarrow \exists z\ x + Sz = y$
$x \leq y : \Leftrightarrow \exists z\ x + z = y$

As we might expect, these predicates may be defined in terms of each other:

15) $x < y \longleftrightarrow x \leq y \ \wedge \neg x = y$
$\rightarrow$: Let $x + Sz = y$, then there exists an u, such that $x + u = y$, hence $x \leq y$. Let $x = y$ then $Sz + x = x$, and $Sz = \underline{0}$ (by 11)

$\leftarrow$: Assume $x + z = y$, if $z = \underline{0}$ then $x = y$, contrary to the assumption. Ergo $z = Su$, in other words: $x + Su = y$, which by definition is equivalent to $x < y$

16) $x \leq y \longleftrightarrow x < y \vee x = y$
$\rightarrow$: Let $x + z = y$. If $z = \underline{0}$ then $x = y$ if $z = Su$ then $x < y$ (1)
$\leftarrow$: If $x < y$ then $x \leq y$. If $x = y$ then $x + \underline{0} = y$, and hence $x \leq y$

17) $\leq$ is a weak partial order (wpo), i.e.
$x \leq x$ (reflexivity)
$(x \leq y \ \wedge\ y \leq x) \rightarrow x = y$ (Antisymmetry)
$(x \leq y \ \wedge\ y \leq z) \rightarrow x \leq z$ (Transitivity)

a.) $x \leq x$
$x + \underline{0} = x$
b.) $x \leq y \wedge y \leq z \rightarrow x \leq z$
Assume $x + u = y, y + v = z$, then $(x + u) + v = z$, and hence (3,4) $x + (u + v) = z$, yielding $x \leq z$
c.) $x \leq y \wedge y \leq x \rightarrow x = y$
Assume $x + u = y$ and $y + v = x$ then $(x + u) + v = x$. This yields (3,4) $x + (u + v) = x$, hence (3) $(u + v) + x = \underline{0} + x$, ergo $u + v = \underline{0}$ (11). We conclude (10) that $u = \underline{0}, v = \underline{0}$ hence $x = y$.

18) $<$ is strong partial order (spo):
$\neg x < x$
$(x < y \ \wedge\ y < x) \rightarrow x < y$
$(s < y \ \wedge\ y < z) \rightarrow x < x$

An exercise for the reader. Hint: Use the previous result.

19) $x < y \longleftrightarrow Sx \leq y$ (2)
Immediate from 2. $x + Su = y \longleftrightarrow Sx + u = y$

20) $x < y \ \vee\ x = y \ \vee\ y < x$
– $x < \underline{0} \vee x = \underline{0} \vee \underline{0} < x$, since: (3) $\underline{0} + x = x$, hence $\underline{0} \leq x$, ergo (16) $x = \underline{0} \vee \underline{0} < x$
– Assume $x < y \vee x = y \vee y < x$. If $x < y$, then (19) $x + Su = y$, so $x + SSu = Sy$, hence $x < Sy$. If $x = y$, then $x + \underline{1} = Sx = Sy$, and therefore $x < Sy$. If $y < x$, then

$Sy \leq x$ i.e. $Sy < x \vee x = Sy$.

21) $x < Sy \longleftrightarrow x \leq y$

$x + u = y \longleftrightarrow S(x + u) = Sy \longleftrightarrow x + Su = Sy$

22) $x < y \longleftrightarrow x+z < y+z$

$x + Su = y \longleftrightarrow (x + Su) + z = y + z \longleftrightarrow (x + z) + Su = y + z$

23) $x < y \longleftrightarrow x \cdot Sz < y \cdot Sz$

$' \rightarrow '$: Assume $x + Su = y$ then $(x + Su) \cdot Sz = y \cdot Sz$, hence: (7)
$xSz + SuSz = y \cdot Sz$ or
$xSz + (Su \cdot z + Su) = y \cdot Sz$ or
$xSz + S(Su \cdot z + u) = y \cdot Sz$, and we can conclude:
$x \cdot Sz < y \cdot Sz$

$' \leftarrow '$: Assume $\neg(x < y)$. Then, by result 20, $x = y$ or
$y < x$. Both cases lead to a contradiction: If $x = y$,
$xSz = ySz$. If $y < x$, then $ySz < xSz$ by the first part.

We focus our attention now on basic principles that are frequently used in proving other results. Among them are transfinite induction, the minimum principle, the maximum principle and the less well-known (but equally important) collection principle.
Notation: (if x does not occur free in t)
$\exists x \lhd t\ Ax : \Leftrightarrow \exists x(x < t \wedge Ax)$
$\forall x \lhd t\ Ax : \Leftrightarrow \forall x(x < t \rightarrow Ax)$
24) TRANSFINITE INDUCTION
$\forall x(\forall y < x\ Ay \rightarrow Ax) \rightarrow \forall x Ax$
(Exercise: Show that TI is weaker than ordinary induction.)

Assume $\forall x(\forall y < x Ay \rightarrow Ax)$. We use induction on z in $\forall x < z\ Ax$.
- $\forall x < \underline{0}\ Ax$
- $\forall x < Sz\ Ax \longleftrightarrow \forall x(x < z \vee x = z) \rightarrow Ax$
  $\longleftrightarrow \forall x(x < z \rightarrow Ax) \wedge \forall x(x = z \rightarrow Ax)$
  $\longleftrightarrow (\forall x < z Ax) \wedge Az$
  $\longleftrightarrow^{IH, ASS} T$

We get: $\forall z(\forall x < z Ax)$. For a particular x_0:
$\forall x < Sx_0\ Ax$, $x_0 < Sx_0$, ergo Ax_0.
Hence $\forall x Ax$

Define:
$\exists\ \mu x\ Bx : \Leftrightarrow \exists x(Bx \wedge \forall y < x\ \neg By)$

25) MINIMUM PRINCIPLE
$\exists x\ Bx \rightarrow \exists \mu x\ Bx$
An exercise for the reader.
(Hint: Take $Ax = \neg Bx$ in 24 and use contraposition.)

Define:
$\exists \lambda x < y\ Ax : \Leftrightarrow \exists x < y\ (Ax \wedge \forall z < y\ (Az \rightarrow z \leq x)$
26) MAXIMUM PRINCIPLE
$\exists x < y\ Ax \rightarrow \exists \lambda x < y\ Ax$

Assume $\exists x < y\ Ax$.
Take $Cx := \forall z < y(z > x \rightarrow \neg Az)$. $\exists x < y\ Cx$, because $\exists x\ x < y$,
hence $y = Su$. We show: Cu. If $u < z < y$ then $u + Sv = z$, $z + Sw = y$,
yielding $(u + Sv) + Sw = u + (Sv + Sw) = u + SS(v + w) = u + S\underline{0}$. We find
$SS(v + w) = S\underline{0}$, and this leads to a contradiction: $S(v + w) = \underline{0}$.

It follows that $\exists \mu x\ Cx$, say x_0. Clearly $x_0 < y$ Claim: Ax_0.
Suppose that $\neg Ax_0$. We distinguish to cases (cf. 1.):
Case 1: $x_0 = Su_0$. Take a z with $z < y$ and $z > u_0$.
$z > u_0 \leftrightarrow z \geq Su_0 \leftrightarrow z > Su_0 \vee z = Su_0$.
If $z > Su_0 = x_0$ then $\neg Az$. But $u_0 < x_0$.
Contradiction.
If $z = Su_0$ then $\neg Az$ too. Contradiction.
Case 2: $x_0 = \underline{0}$. Assume $\neg Ax_0$ then
$\forall z < y(\underline{0} < z \rightarrow \neg Az \wedge \underline{0} = z \rightarrow \neg Az)$, in other words:
$(\forall z < y\ \underline{0} \leq z \rightarrow \neg Az)$. It follows that: $\forall z < y\ \neg Az$.
Contradiction.
(end proof of claim)
This concludes the proof of the maximum principle.

27) PA - Induction + TI + $\forall x\ (x = \underline{0} \vee \exists y\ (x = Sy)) \vdash$ *Induction*
Assume Axioms 1-6, TI, $\forall x\ (x = \underline{0} \vee \exists y\ (x = Sy))$. In addition,
asume that: $A\underline{0} \wedge \forall x(Ax \rightarrow ASx)$. It is sufficient to prove:
$\forall y((\forall x < yAx) \rightarrow Ay)$. Let $\forall x < yAx$. If $y = \underline{0}\ A\underline{0}$.
If $y = Su$, $u < y$, it follows that Au hence ASu, i.e. Ay.

28) $((Bx \wedge \forall y < x\ \neg By) \wedge (Bz \wedge \forall y < z\ \neg By)) \rightarrow x = z$
An exercise for the reader.
(Hint: use fact 19).

29) COLLECTION PRINCIPLE
$\forall x < y\ \exists z\ Axz \leftrightarrow \exists u\ \forall x < y\ \exists z < u\ Axz$
$\leftarrow$: trivial.
$\rightarrow$: induction on y.
- $\underline{0}$ trivial
- $\forall x < Sy\ \exists zAxz$, then
  $\forall x < y\ \exists zAxz$ and $\exists zAyz$, so
  $\exists u \forall x < y \exists z < u\ Axz$ and $\exists z_0\ Ayz_0$.
  Take such u, z_0, and $v := max(u, z_0 + \underline{1})$.
  then $\forall x < y\ \exists z < v\ Axz, \exists z_0 < v\ Ayz_0$.
  $\forall x < Sy\ \exists z < v\ Axz$.

Properties of division
DEFINITION:
$a \mid b : \Leftrightarrow \exists\ x\ a \cdot x = b$
p is prime: $\Leftrightarrow \neg p = \underline{0} \wedge \neg p = \underline{1} \wedge \forall x\ (x \mid p \rightarrow (x = \underline{1} \vee x = p))$
30) $\underline{1} \mid b$
Simply because $\underline{1} \cdot b = b$

31) $a \mid \underline{0}$
$a \cdot \underline{0} = \underline{0}$

The following three results prove that | is a weak partial order.

32) $a \mid a$
$a \cdot \underline{1} = a$

33) $(a \mid b \wedge b \mid c) \rightarrow a \mid c$

$a \cdot x = b, b \cdot y = c \rightarrow (ax)y = a(xy) = c$

34) $(a \mid b \wedge b \mid a) \rightarrow a = b$
$ax = b, by = a \rightarrow a(xy) = a = a \cdot \underline{1}$
Now, distinguish two cases:
$a \neq \underline{0}$: $xy = \underline{1}$ hence $x = \underline{1}, y = \underline{1}$
$a = \underline{0}$: $\underline{0} \cdot x = b$, hence $b = \underline{0}$

35) $a \mid b \rightarrow c \cdot a \mid c \cdot b$
$ax = b \rightarrow cax = cb$

36) $(\neg c = \underline{0} \wedge c \cdot a \mid c \cdot b) \rightarrow a \mid b$
$cax = cb \wedge c \neq \underline{0} \rightarrow ax = b$

37) $(a \mid b \wedge \neg b = \underline{0}) \rightarrow a \leq b$
$a \cdot x = b$. Suppose that $x = \underline{0}$. Then $b = \underline{0}$,
contrary to the assumption. Hence $x \neq \underline{0}$. But then:
$\underline{1} \leq x$, because if $x = Sy$ then $\underline{1} + y = Sy$.
Hence $a \cdot \underline{1} < a \cdot x \mid b$.

38) $(a \neq \underline{0} \wedge a \neq \underline{1}) \rightarrow \exists p$ (p is prime $\wedge\, p \mid a$)
Suppose $a \neq \underline{0} a \neq \underline{1}$. Then:
$a \mid a \rightarrow \exists x\ x \neq \underline{0} \wedge x \neq \underline{1} \wedge x \mid a$
$\rightarrow \exists \mu x\ x \neq \underline{0} \wedge x \neq \underline{1} \wedge x \mid a$

Call this smallest divisor x of a: p.
Claim: p is prime. Suppose $y \mid p$, then $y \mid a$ (by result 33).
$p \neq \underline{0}$, hence $y \leq p$, and $y \mid a$ and $y \neq \underline{0}$. We show: $y = p$
or $y = \underline{1}$. If $y = p$, we are ready, if $y \neq p$, it follows that $y < p$,
and hence: $y = \underline{0} \vee y = \underline{1} \vee \neg(y \mid a)$, ergo $y = \underline{1}$.

39) $p \mid (p \cdot q + r) \rightarrow p \mid r$
$p \mid p \cdot q + r$, so by definition $px = pq + r$.
It follows that $q \leq x$ (result 23).
Let $q + z = x$, then $p(q + z) = pq + r$, and
by result 6 & 12 (commutativity of $\cdot$ and the
distributive property) we find $pq + pz = pq + r$.
Applying result 6, we get $pz = r$, which is equivalent
to the desired proposition.

40) $\forall x\ \exists p \geq x$ p is prime.
The statement will follow from:
$\forall x\ \exists y \neq \underline{0}\ \forall z < x\ z \neq \underline{0} \rightarrow z \mid y$
We will prove this by induction on x
- $x = \underline{0}$. Trivial.
- $x = Su$.
  Induction Hypothesis:
  $\exists v \neq \underline{0}$ with $\forall z < u\ z \neq \underline{0} \rightarrow z \mid v$
  Now, take $y := v \cdot u$, if $u \neq \underline{0}$, $\underline{1}$ if $u = \underline{0}$.
  y satisfies the requirement:
  Take $z < Su$. Distinguish two cases:
  $z = u$: $z \mid v \cdot u$. Trivial.
  $z < u$: $z \mid v \cdot u$, by the induction hypothesis.

Consider x and y ($y \neq \underline{0}$) such that for all $z < x$:
$z \neq \underline{0} \rightarrow z \mid y$. y exists for an arbitrary x by the proposition above. Let p be a prime factor of $(y + 1)$.
Such a factor exists, while from our assumptions, we may deduce that $y + \underline{1} \neq \underline{0}$, and $y + \underline{1} \neq \underline{1}$.
Assume that $p \mid y$. Then, by the previous result $p \mid \underline{1}$. This is contradictory to the assumption that p is prime. Hence p does not divide y. We conclude that $p \geq x$.

Define:
$(\exists ! x_1, \ldots, x_k \; A(x_1, \cdots, x_k) : \Leftrightarrow$
$\exists x_1 \exists x_2 \cdots \exists x_k \; (A(x_1, \cdots, x_k) \wedge \forall y_1 \forall y_2 \cdots \forall y_k$
$(A(y_1, \cdots, y_k) \rightarrow (x_1 = y_1 \wedge \cdots \wedge x_k = y_k))$

41) EUCLIDES' ALGORITHM
$\forall a \; \forall b \; (a \neq \underline{0} \wedge b \neq \underline{0}) \rightarrow \exists ! n,r \; (a = n \cdot b + r \wedge r < b)$

First, we prove existence of n and r. Uniqueness will be dealt with thereafter.

Existence:
Since $\underline{0} \cdot b = \underline{0} \leq a$, we know that $\exists x < a + \underline{1} \;\; x \cdot b \leq a$
By the maximum principle,
$\exists \lambda x \; < a + \underline{1} \; (x \cdot b \leq a)$
Call this largest x n. By definition of $\leq$, we have
$n \cdot b + r = a$ for some r.
Assume that $r \geq b$, i.e. $r = b + r'$. Then
$nb + r = nb + b + r' = (n + \underline{1})b + r' = a$
$n + \underline{1} \geq n$, contrary to the fact that n is the largest x such that $x \cdot b \leq a$.
We conclude that $r < b$.

Uniqueness:
Assume that $a = nb + r, a = n'b + r', \;\; r,r' < b$.
If $n < n'$, then $n' = n + Sz$. Substitution yields:
$nb + r = (n + Sz)b + r' = nb + Szb + r'$
This leads us to the fact that
$r = Szb + r' = b + (bz + r')$, and $r \geq b$.
This is contradictory to the assumption that $b < r$, hence $n' \leq n$. By symmetry, we conclude that $n \leq n'$ too. But then $n = n'$, and since $nb + r = n'b + r'$, we may conclude that $r = r'$.
This completes the proof of the algorithm.

42) $(p$ is prime $\wedge p \mid a \cdot b) \rightarrow (p \mid a \vee p \mid b)$
An exercise for the reader.

Above, we saw that | defines a weak partial order. This together with the next two results, yields the fact that $<|PA|, | >$ is a lattice. A lattice is a weakly partially ordered set in which every two elements have a greatest lower bound and a least upperbound and which has a bottom and a top.

DEFINITION:
$\text{LCM}(a,b) = c : \Leftrightarrow a \mid c \wedge b \mid c \wedge$
$\forall d \; ((a \mid d \wedge b \mid d) \rightarrow c \mid d)$

43) LCM defines a function
UNIQUENESS
$LCM(a,b) = c \wedge LCM(a,b) = c' \rightarrow c = c'$
EXISTENCE
$\exists c \;\; LCM(a,b) = c$

Uniqueness:

Suppose $c = LCM(a,b)$ and $c' = LCM(a,b)$. It follows that $c \mid c'$ and $c' \mid c$. By result 34, $c = c'$.

Existence:
The proof is by TI on a:
We distinguish the following cases:
CASE I:
$a = \underline{0}$. Take $c := \underline{0}$. Clearly, this c satisfies our requirements.
CASE II:
$a = \underline{1}$. Take $c := b$.
CASE III:
$a \neq \underline{0}$, $a \neq \underline{1}$. Then a has a prime factor q.

Case IIIa: $q \mid b$
$q \mid b$, in other words: $b = qb'$. q is a prime factor of a:
$a = qa'$.
Take $c := q \cdot LCM(a',b') = qc'$. Note that c' exists by the induction hypothesis.
$a' \mid c'$, hence $qa' \mid qc'$ [1]

$b' \mid c'$, hence $qb' \mid qc'$ [2]

Suppose that $a \mid d$ and $b \mid d$. Then $qa' \mid d$, $qb' \mid d$. It follows that
$d = x \cdot q \cdot a'$ and $d = y \cdot q \cdot b'$.
$x \cdot q \cdot a' = y \cdot q \cdot b'$, since $q \neq \underline{0}$:
$x \cdot a' = y \cdot b'$. Call this number d'. $d = qd'$,
$a' \mid d'$ and $b' \mid d'$. $c' = LCM(a',b')$, so $c' \mid d'$. We conclude that $qc' \mid qd'$ [3].

Combining [1], [2], [3], we conclude that c satisfies the requirements.

Case IIIb: $\neg\, q \mid b$
Take $c := q \cdot LCM(a',b) = q \cdot c'$, where $a = qa'$
$a' \mid c'$, hence $qa' \mid qc'$ [1]

$b \mid c'$, hence $b \mid qc'$ [2]

Suppose that $a \mid d$ and $b \mid d$. Again:
$d = x \cdot qa' \rightarrow d = y \cdot qb'$ Take $d' = xa' = yb' (q \neq \underline{0})$
$a' \mid d'$. $b \mid qd' = d$, hence $bz = qd'$ for some z.
By the previous result (42), we get:
$q \mid b$ or $q \mid z$.
We assumed that $\neg q \mid b$, so $q \mid z$. Substituting this in $bz = qd'$, one finds: $bqz' = qd'$, hence
$b \cdot z' = d'$, and $b \mid d'$.
Now, we apply the fact that $c' = LCM(a',b)$ to find that $c' \mid d'$, and hence $qc' \mid qd'$. In other words:

$c \mid d$ [3].

Combining [1], [2], [3], we conclude that c satisfies the requirements.

44) The case of GCD is left to the industrious reader.

DEFINITION:
Let p be a prime, define
d is a power of $p :\Leftrightarrow d \neq \underline{0} \ \wedge \ \forall x \ (x \mid d \rightarrow x = \underline{1} \ \vee \ p \mid x)$.

Show that the definition doesn't work if p is not a prime.

45) (p is prime $\wedge$ d is a power of p $\wedge$ d' is a power of p) $\rightarrow$ $d \cdot d'$ is a power of p

Proof:

$d \cdot d' \neq 0$. Suppose $x \mid d \cdot d'$ and $x \neq \underline{1}$
Suppose q is the smallest prime such that $q \mid x$. From 44 follows $q \mid d \vee q \mid d'$. Because $q \neq \underline{1}$, $p \mid q$ follows, hence $p = q$.
Conclude $p \mid x$.

46) (p is prime $\wedge$ d is a power of p $\wedge$ d' is a power of p) $\rightarrow (d \mid d' \longleftrightarrow d \leq d')$

Proof:

$\rightarrow$: is already done.
$\leftarrow$: Suppose $d \leq d'$. $\underline{1} \mid d$, $\underline{1} \mid d'$, so a power of p exists, which divides both. Such powers are $\leq d$. Let d_0 be the largest. If $d_0 = d$ then we are finished.
Else $d = d_0 \cdot e$, where $e \neq \underline{1}$, hence $p \mid e$.
Therefore $d = d_0 \cdot p \cdot e'$. Also $d' = d_0 \cdot f$, where $f \neq \underline{1}$, hence $d' = d_0 \cdot p \cdot f'$.
It follows that $d_0 \cdot p \mid d$ and $d_0 \cdot p \mid d'$ and $d_0 \cdot p > d_0 \cdot 1 = d_0$. Contradiction.

47) $\forall x \ \exists \ d > x$ d is a power of p

Proof:

With induction:
- $\underline{0}$ is trivial.
- Induction hypothesis: $d > n$.
  Then $p \cdot d \geq 2 \cdot d = d + d \geq d + 1 > n + 1$.

DEFINITION:
$\eta(x) := \mu d \ (d$ is a power of $p \ \wedge \ d > x)$

48) $x \neq \underline{0} \rightarrow (d = \eta(x) \longleftrightarrow d$ is a power of $p \ \wedge \ d > x \ \wedge \ d \leq p \cdot x$

Proof: An exercise for the reader.

2. Concatenation

To prove Gödel's Incompleteness Theorems, we will have to code elementary syntax in arithmetic. In this section we will carry out a first step of this task: we define an arithmetical operation with all the properties of concatenation. Our concatenation operation is a minor modification of the one found in Boolos & Jeffrey.

DEFINITION:
$a * b := a \cdot \eta(b) + b$

49) $\eta(a * b) = \eta(a) \cdot \eta(b)$

Proof:

1. $\eta(a) \cdot \eta(b)$ is a power of p
2. $\eta(a) \cdot \eta(b) \geq (a + 1) \cdot \eta(b) = a \cdot \eta(b) + \eta(b) >$
   $a \cdot \eta(b) + b$.
3. $p \cdot (a \cdot \eta(b) + b) \geq p \cdot a \cdot \eta(b) \geq \eta(a) \cdot \eta(b)$ (cf.48)

The case $a, b = 0$ is trivial.

50) $(a * b) * c = a * (b * c)$

Proof:

$$(a \cdot \eta(b) + b) \cdot \eta(c) + c =$$
$$a \cdot \eta(b) \cdot \eta(c) + b \cdot \eta(c) + c =$$
$$a \cdot \eta(b * c) + (b * c) =$$
$$a * (b * c).$$

51) $a * c = b * c \rightarrow a = b$

Proof: An exercise for the reader.

DEFINITION:
$b \subseteq a :\Leftrightarrow \exists c \, \exists c' \, (a = c * b * c')$
$b \subseteq_i a :\Leftrightarrow \exists c' \, (a = b * c')$
$b \subseteq_e a :\Leftrightarrow \exists c \, (a = c * b)$

52) $(b \subseteq_e a \wedge c \subseteq_e a) \rightarrow (b \subseteq_e c \vee c \subseteq_e b)$

Proof:

$a = d * b = e * c$, so $d \cdot \eta(b) + b = e \cdot \eta(c) + c$.
Suppose $b \leq c$, say $b + x = c$,
then $d \cdot \eta(b) = e \cdot \eta(c) + x$.
$\eta(b) \leq \eta(c)$, hence $\eta(b) \mid \eta(c)$, hence $\eta(b) \mid x$ (cf. 46).
Therefore $c = x' \cdot \eta(b) + b = x' * b$.

53) $a * b = a * c \rightarrow b = c$

Proof:

Let $b \subseteq_e c$, i.e. $c = b' * b$.
Then $a * b = a * b' * b$, hence
$a = a * b' = a \cdot \eta(b') + b'$.
Therefore $b' = 0$.

54) $(b \subseteq_i a \ \wedge \ c \subseteq_i a) \rightarrow (b \subseteq_i c \ \vee \ c \subseteq_i b)$

Proof:

$a = b * b' = c * c'$, now $b' \subseteq_e a$, $c' \subseteq_e a$.
It follows that $b' \subseteq_e c' \vee c' \subseteq_e b'$.
Let $b' \subseteq_e c'$ then $c' = b'' * b'$.
Hence $b * b' = c * b'' * b'$. With 51 it
follows that $b = c * b''$.
Therefore $c \subseteq_i b$. The case $c' \subseteq_e b'$ is similar.

55) $a \subseteq c * b \rightarrow (a \subseteq c \ \vee \ a \subseteq b \ \vee \ \exists c' \subseteq_e c \ \exists b' \subseteq_i b \ (a = c' * b'))$

Proof:

Another way to formulate this is:
$a \subseteq c * b \rightarrow a \subseteq c \vee a \subseteq b \vee (c = c' * c''$,
$b = b' * b'' \wedge a = c'' * b')$
Suppose
$c * b = x * a * y$.
Case 1.
$c \subseteq_i x$, so $c * b = c * x' * a * y$, hence
$b = x' * a * y$, i.e. $a \subseteq b$
Case 2.
$x \subseteq_i c$, so $x * c' * b = x * a * y$, hence
$c' * b = a * y$

Case 2.1
$a \subseteq_i c'$: Then $c = x * a * c''$, hence
$a \subseteq c$.
Case 2.2
$c' \subseteq_i a$: Then $c' * b = c' * a' * y$, hence
$b = a' * y$
Therefore $c = x * c'$, $b = a' * y$ and $a = c' * a'$

DEFINITION:
e is a p-atom $:\Leftrightarrow e \neq \underline{0} \ \wedge \ \forall f \ \forall g \ (e = f * g \rightarrow (f = \underline{0} \ \vee \ g = \underline{0}))$

56) $(e$ is a p-atom $\wedge \ e \subseteq a * b) \rightarrow (e \subseteq a \ \vee \ e \subseteq b)$

Proof: Left to the reader.

57) $(d$ is a power of $p \ \wedge \ a \neq \underline{0}) \rightarrow \ \eta(d \cdot a) = d \cdot \eta(a)$

Proof:

1.$d \cdot \eta(a)$ is a power of p.
2.$d \cdot \eta(a) > d \cdot a$.
3.$d \cdot \eta(a) \leq d \cdot (p \cdot a) = p \cdot (d \cdot a)$.

58) q is a p-atom $\longleftrightarrow \ \exists m \ \exists d \ (d$ is a power of $p \ \wedge \ \underline{0} < m < p \ \wedge \ q = m \cdot d)$

Proof:

$\leftarrow$ Suppose $q = a * b$, $q = m \cdot d$, d is a power of p and

$\underline{0} < m < p$.
Then $m \cdot d = a \cdot \eta(b) + b$. Suppose $b \neq \underline{0}$, then
$d \mid a \cdot \eta(b) + b$ and $\neg\, \eta(b) \mid a \cdot \eta(b) + b$, hence
$\neg\, \eta(b) \mid d$, hence $d \mid \eta(b)$, hence $d \mid b$.
It follows that $m \cdot d = a \cdot e \cdot d + b' \cdot d$, i.e.
$m = a \cdot e + b'$.
Now $b' \neq \underline{0}$, e is a power of p, $\underline{0} < m < p$.
Therefore $a = \underline{0}$ or $e = \underline{1}$. In the first case the proof is completed. If $e = \underline{1}$ then $\eta(b) = d$. Contradiction.
$\rightarrow$ Suppose q is a p-atom.
Let d_0 be the largest power of p, which divides q (and $\leq q$).
Then $q = m_0 \cdot d_0$ and $\neg\, p \mid m_0$.
Suppose $m_0 \geq p$, hence $m_0 > p$.
It follows that $m_0 = b \cdot p + c$, $b \neq \underline{0}$, $\underline{0} < c < p$, hence
$q = (b \cdot p + c) \cdot d = b \cdot p \cdot d + c \cdot d$. Now
$\eta(c \cdot d) = \eta(c) \cdot d = p \cdot d$, hence $q = b * c \cdot d$.
Contradiction. Conclude $m_0 < p$.

3. Sequences

The most important thing we need in order to obtain the possibility to talk syntax in arithmetic is a coding of sequences.

Gödel in his original paper used the Chinese Remainder Theorem to code sequences, we will do it using our concatenation operation. The two characteristic properties of sequences are:

i) they have a length.

ii) there is a projection function pr such that: for $i <$ length(s) pr(s,i) is the i-th number in s. Instead of pr(s,i), we write $(s)_i$.

The obvious idea to code a sequence $\langle n_0,\ldots,n_{l-1}\rangle$ is to pick some number k to code the comma and take: $n_0 * k * \cdots * k * n_{l-1}$.

There are two problems with this proposal, one trivial and one difficult.

The trivial one is that there is no obvious way to count to the i-th member of the coded sequences to obtain $(s)_i$. We could do this if we had recursion, but to show that we can define functions by recursion in arithmetic we will precisely need sequences. The solution to the trivial problem is to put $\langle i,n_i\rangle$ or better : a code of $\langle i,n_i\rangle$ into the coded sequences rather then n_i itself.

The difficult problem has to do with the comma: what if e.g. $n_0 = m * k * m'$. How can we distinguish $\langle n_0,n_1,\ldots\rangle$ from $\langle m,m',n_1,\ldots\rangle$? One idea is to use for each sequence a different comma: then to give a sequence is both to give its code and the code of the comma. This would indeed solve the problem, but it would create a new problem: we will have to prolong sequences i.e. to go from $\langle n_0,\ldots,n_{l-1}\rangle$ to $\langle n_0,\ldots,n_{l-1},m\rangle$ for arbitrary m. If the code of the comma coming with the code of the original sequence is k and $k \subseteq m$, then we will have to change the comma in the whole original sequence to a new one, not part of m. The only way we can see to do that is by recursion...

A second idea is to code our numbers $n_0,\ldots,n_{l-1}$ first into $\hat{n}_0,\ldots,\hat{n}_{l-1}$, in such a way that our chosen comma k does not occur in $\hat{n}_0,\ldots,\hat{n}_{l-1}$. The problem is now to find an appropriate function ($\hat{}$) that can be defined with the apparatus at hand (We couldn't find one).

The idea we will use here is the following: we will make codes of sequences with growing commas: the commas will always be so big that they can be distinguished from the numbers in the sequence.

CONVENTIONS

- We use d, d' always for powers of p without mentioning this explicitly.
- We write e.g. xdw instead of $x * d * w$

DEFINITION:

$w \in r :\Leftrightarrow \exists x\, \exists d(\, d$ is a power of p $\wedge$
$r = x * d * w \wedge$
$d > x \wedge d > w)$
$\vee$
$\exists x\ \exists y\ \exists d\ \exists d'\ (\, d$ is a power of p $\wedge$
d' is a power of p $\wedge$
$r = x * d * w * d' * y\ \wedge$
$d > x \wedge d > w \wedge d' > xdw)$

It is intended that w is a number occurring in r, where the d, d' function as commas.

59) $rdw = r'd'w' \wedge d > w \wedge d' > r' \rightarrow$
$((r = r' \wedge d = d' \wedge w = w') \vee$
$(r = r'd'x' \wedge w' = x'dw))$

We distinguish two cases:

a) $r'd' \subset_i r$ then $r = r'd'x'$ and $r'd'x'dw = r'd'w'$ hence $w' = x'dw$
b) $r \subset_i r'd'$, but $r \neq r'd'$ then $ru = r'd'$, $u \neq \underline{0}$. It follows that $d' \subset_e u$. Hence: $ru'd' = r'd'$. Now $rdw = ru'd'w'$
*CASE b*1) $u' = \underline{0}$ then $d = d', r = r', w = w'$
*CASE b*2) $u' \neq \underline{0}$ then $u' = du''$ and hence
$r' = ru' = rdu''$, so $d' > d$. Also, $w = u''d'w'$ yields $d > d'$
Contradiction.

60) $(d > r \wedge d > w) \rightarrow (w' \in rdw \longleftrightarrow (w' \in r \vee w' = w))$

Assume that $(d > r \wedge d > w)$. In addition, assume that $w' \in rdw$
Again, we can distinguish two cases:
a) $r'd'w' = rdw$ then either $(r' = r, d' = d, w' = w)$ or $(r = r'd'x'$ and $w' = x'dw)$, hence $d > d'$, $d' > d$. Contradiction. We conclude that $w = w'$.
b) $rdw = r'd'w'd''r''$. By 59 (substitute d'' for d') it follows that either $r = r'd'w'$ and hence $w' \in r$, or $r = r'd'w'd''x'$ and $w' \in r$ too. $w = w'$

DEFINITION

a) $a \circ b = a * \eta(max(a, b)) * b$
b) $Seq(r) :\Leftrightarrow ((k \circ w \in r \wedge k \circ w' \in r) \rightarrow w = w')$
$\wedge$
$((k \circ w \in r \wedge l < k) \rightarrow \exists w'\ l \circ w' \in r)$
c) $length(r) := \lambda k <_{r+\underline{1}}$
$k = \underline{0} \vee \exists k'(k = Sk' \wedge k' \circ w \in r))$
d) $(r)_i := \mu w <_{r+\underline{1}} i \circ w \in r$

61) $(Seq(r) \wedge length(r) = k) \rightarrow Seq(r \circ (k \circ w))$

Assume: $k' \circ w' \in r \circ (k \circ w)$
$k' \circ w'' \in r \circ (k \circ w)$
i) $k' \circ w' = k' \circ w'' = k \circ w$ then $k' = k$, $w' = w'' = w$.
ii) $k' \circ w' \in r$, then $k' < r + 1$ ergo $k < Length(r)$. It follows that $k' \circ w' \in r$. (Otherwise: $k' \circ w' = k \circ w$, hence $k = k'$. Contradiction.)
$Seq(r)$ hence $w' = w''$.

iii)

$k' \circ w'' \in r$. Analogous to the previous case.

Assume $k' \circ w' \in r \circ (k \circ w)$.

If $k' \circ w' \in r$, we are ready for $l < k'$.

If $k' \circ w' = k \circ w$ then $k' = Length(r)$, so for $l < k$ there exists an w'' with $l \circ w'' \in r$.

62) $(Seq(r) \wedge length(r) = k \rightarrow$

$$r \circ (k \circ w))_i = \begin{cases} (r)_i & \text{if } i < k \\ w & \text{if } i = k \\ (r \circ (k \circ w)) \dot{-} \underline{1} & \textit{otherwise} \end{cases}$$

An exercise for the reader

EXAMPLE

Let $p := 2$. We use (61) to obtain a code for $\langle \underline{0},\underline{0},\underline{0} \rangle$.

$0 \circ 0 = 0 * 1 * 0 = 1$
$1 \circ 0 = 1 * 2 * 0 = 6$
$2 \circ 0 = 2 * 4 * 0 = 20$
$0 \circ (0 \circ 0) = 0 \circ 1 = 0 * 2 * 1 = 5$
$5 \circ (1 \circ 0) = 5 \circ 6 = 5 * 8 * 6 = 710$
$710 \circ (2 \circ 0) = 710 \circ 20 = 710 * 1024 * 20 = 46563348$

EXAMPLE

Let $p := 3$.

Which sequence is coded by 392912083? We first rewrite the number as a concatenation of atoms. This is best done by first finding the last atom, then the last atom of the string before it, etc. The last atom of a number is the rest of division by the smallest 3-power not dividing it.

$393912083 = 130970694 \cdot 3 + 1 = 130970694 * 1$
$130970694 = 14552299 \cdot 9 + 3 = 14552299 * 3$
$14552299 = 4850766 \cdot 3 + 1 = 4850766 * 1$
$4850766 = 739 \cdot 3^8 + 3^7 = 739 * 3^7$
$739 = 246 \cdot 3 + 1 = 246 * 1$
$246 = 27 \cdot 3^2 + 3 = 27 * 3$
$27 = 3^3$.

Hence

$392912083 = 3^3 * 3 * 1 * 3^7 * 1 * 3 * 1$

Finally it is easy to see that this codes: $\langle \underline{1},\underline{1} \rangle$.

4. Extensions of theories

It is time to reflect on what we are doing. The sentence 'All sequences have a code' can't be expressed in PA. But ' $\forall r \forall n$ ($seq(r) \rightarrow$ $seq(r \circ n)$) ' can be written down in PA, because of the way

we have enriched the vocubulary of PA. The latter sentence will turn out to be sufficient for our goals. Intuitively the process of enriching a language will not introduce new theorems: any theorem proved in the enriched language can be translated back to the original language and the proof of a theorem can be translated to a proof of the corresponding translated theorem. This intuition should be sufficient to inspire trust in what we are doing. But we really need to know more (how complex will the formulas become when you translate them back to PA ?) about the translation (for future use). Hence we will give an analysis of the process of extending the language.

4.1. Definition

$T \ll U :\Leftrightarrow U$ is a definitional extension of T
$:\Leftrightarrow L_U$ extends L_T with relation symbols $R_0, \ldots, R_{k-1}$ and function symbols $F_0, \ldots, F_{l-1}$ and $U = T + \forall \vec{x}\ (R_i(\vec{x}) \longleftrightarrow B_i(\vec{x})) + \forall \vec{x},y\ (F_j(\vec{x}) = y \longleftrightarrow A_j(\vec{x},y))$
where $B_i, A_j \in L_T$ and $T \vdash \forall \vec{x} \exists !y\ A_j(\vec{x},y)$

4.2. Definition

Suppose L_U extends L_T, we say that U is conservative over T if $\forall A \in L_T\ (U \vdash A \Rightarrow T \vdash A)$

4.3. Theorem

If $T \ll U$ then U is conservative over T.

Proof:

The proof is trivial, but tedious and rather long. Consider T,U with $T \ll U$. To simplify a bit, suppose L_T has only one relation symbol $=$ and binary functionsymbol G and suppose the only new symbol in L_U is an unary functionsymbol F, where $U \vdash \forall x,y\ (F(x) = y \longleftrightarrow A(x,y))$ and $T \vdash \forall x \exists !y\ A(x,y)$. We define a translation from L_U to L_T as follows:

(i) $(c = x)^* := (c = x)$

(ii) for $x \neq y$: $(y = x)^* := (y = x)$

(iii) for $x \notin FV(G(t_1,t_2))$:
$(G(t_1,t_2) = x)^* := \exists z_1,z_2\ ((t_1 = z_1)^* \wedge (t_2 = z_2)^* \wedge G(z_1,z_2) = x)$
Here $z_1,z_2 \notin FV(t_1) \cup FV(t_2) \cup \{x\}$

(iv) $(F(t) = x)^* := \exists z\ ((t = z)^* \wedge A(z,x))$
Here $x \notin FV(t)$, $z \notin FV(t) \cup \{x\}$

(v) $(t = t')^* := \exists z\ ((t = z)^* \wedge (t' = z)^*)$
Here $z \notin FV(t) \cup FV(t')$, $t' \notin VARFV(t)$

(vi) $(\ )^*$ commutes with the logical connectives.

Note that for $A \in L_U$: $A^* \in L_T$.

4.3.1. Lemma

(i) $U \vdash B \longleftrightarrow B^*$

(ii) $C \in L_T \Rightarrow T \vdash C \longleftrightarrow C^*$

Proof: Induction on B,C.

4.3.2. Lemma $T \vdash \forall \vec{x} \exists !y\ (t(\vec{x}) = y)^*$

(Where t is in L_U and the free variables of t are among $\vec{x}$ and $y \notin FV(t)$)

Proof: Induction on t.

4.3.3. Lemma

Suppose t is substitutable for x in B, where t,B are in L_U, then (given a fresh variable u):

$T \vdash (B[t/x])^* \longleftrightarrow \exists u\ ((t = u)^* \wedge (B[u/x])^*)$

Proof:

If t is a variable this is easy. Suppose $t \notin VAR$. We induct on the relevant forms of B.

(i) $c = y, y \neq x$ is trivial.

(ii) $z = y, z \neq x, y \neq x$ is also trivial.

(iii) $x = y, y \neq x$: use 4.3.2

$T \vdash (t = y)^* \longleftrightarrow$
$\exists u\ (t = u)^* \wedge (u = y)^* \longleftrightarrow$
$\exists u\ (t = u)^* \wedge u = y$

(iv) $G(t_1(x),t_2(x)) = y$

Here $y \notin FV(t_1(t)) \cup FV(t_2(t)) \cup \{x\}$

Proof:

$T \vdash (\ G(t_1(t),t_2(t)\ ) = y)^* \longleftrightarrow$
$\exists z_1,z_2\ (t_1(t) = z_1)^* \wedge (t_2(t) = z_2)^* \wedge G(z_1,z_2) = y \longleftrightarrow$ (I.H.)

$\exists z_1,z_2\ (\exists v_1\ (\ (t = v_1)^* \wedge (t_1(v_1) = z_1)^*)) \wedge (\exists v_2\ ((t = v_2)^* \wedge (t_2(v_2) = z_2)^*)) \wedge G(z_1, z_2) = y \longleftrightarrow$

$\exists v,z_1,z_2\ ((t = v)^* \wedge (t_1(v) = z_1)^* \wedge (t_2(v) = z_2)^* \wedge G(z_1,z_2) = y \longleftrightarrow$
$\exists v\ (t = v)^* \wedge (\ G(t_1(v),t_2(v)\ ) = y)^*$

v) $F(t_1(x)) = y$ $(y \in FV(t_1(t)) \cup \{x\})$

vi) $t_1(x) = t_2(x)$

$T \vdash (t_1(t) = t_2(t))^* \longleftrightarrow$
$\exists z\ ((t_1(t) = z)^* \wedge (t_2(t) = z)^*) \longleftrightarrow$
$\exists z\ (\exists v_1\ ((t = v_1)^* \wedge (t_1(v_1) = z)^*) \wedge \exists v_2\ ((t = v_2)^* \wedge (t_2(v_2) = z)^*)) \longleftrightarrow$ (4.3.1, 4.3.2)
$\exists v\ \exists z\ ((t = v)^* \wedge (t_1(v) = z)^* \wedge (t_2(v) = z)^*) \longleftrightarrow$
$\exists v\ (t_1(v) = t_2(v))^*$

vii) $\neg B(x)$ An exercise for the reader.

4.3.4. Lemma $\Gamma \vdash_U A \Rightarrow \Gamma^* \vdash_T A^*$

Proof: Induction on proof length.

4.4. Theorem $\ll$ is a partial order. ($W = W'$ if they prove the same theorems)

Proof: the only not trivial case is transitivity. Suppose $T \ll U \ll V$ and let $(\ )^*$ be the translation of L_U to L_T.

Define $V' = T + U$ axioms $+ \forall \vec{x}\ R_i'(\vec{x}) \longleftrightarrow (B_i'(\vec{x}))^* + \forall \vec{x},y\ (F_j'(\vec{x}) = y \longleftrightarrow (A_j'(\vec{x},y))^*)$

Then

$V = V'$

- $L_{V'} = L_V$ is trivial.
- $U \vdash B_i'(\vec{x}) \longleftrightarrow (B_i'(\vec{x}))^*$
  $U \vdash A_j'(\vec{x},y) \longleftrightarrow (A_j'(\vec{x},y))^*$

We want to have more grip on definitional extensions: we especially want to know what kind of formulas we get when we translate back. First we introduce the classes of formulas we are interested in. Why one should be interested in these formulas will become apparent later.

4.5. Definition
Let L_T extend L_{PA} and suppose T extends PA.
a) $\Delta_0(L_T)$ is the smallest class of formulas of L_T s.t.
(i) atomic formulas are in $\Delta_0(L_T)$.
(ii) $\Delta_0(L_T)$ is closed under the propositional connectives.
(iii) $\Delta_0(L_T)$ is closed under $\forall x{<}t$, $\exists x{<}t$ where $x \notin FV(t)$.
b) $\Sigma_1(L_T) := \{\exists \vec{x}\, B \mid B \in \Delta_0(L_T)\}$
c) $\Pi_1(L_T) := \{\forall \vec{x}\, B \mid B \in \Delta_0(L_T)\}$
d) $\Delta(L_T,T) := \{B \in L_T \mid$ there are C,D respectively in $\Sigma_1(L_T)$ and $\Pi_1(L_T)$ s.t. $T \vdash B \leftrightarrow C$ and $T \vdash B \leftrightarrow D\}$

4.6.
Theorem
(i) $\Sigma_1(L_T)$, $\Pi_1(L_T)$ are closed modulo provable equivalence under $\wedge$, $\vee$, $\exists x{<}t$, $\forall x{<}t$.
(ii) $\Delta_0(L_T) \subseteq \Delta(L_T,T)$
(iii) $\Delta(L_T,T)$ is closed under the propositional connectives and bounded quantification.

Proof:
ad (i): with the collection principle.
ad (ii): trivial.
ad (iii):

suppose $B \in \Delta(L_T,T)$ then
$T \vdash B \leftrightarrow A$, $T \vdash B \leftrightarrow \neg A'$, $A,A' \in \Sigma_1$
$T \vdash \forall x{<}t\ B \leftrightarrow \forall x{<}t\ A$
$T \vdash \forall x{<}t\ B \leftrightarrow \forall x{<}t \neg A' \leftrightarrow \neg\ \exists x{<}t\ A'$
etc.

4.7. Definition For PA $\subseteq T,U$; $L_{PA} \subseteq L_T,L_U$
$T \ll_\Delta U :\Leftrightarrow T \ll U$ and the defining formulas B_i are in $\Delta(L_T,T)$ and the defining A_j are in $\Sigma_1(L_T)$.

4.7.1. Fact
If the A_j are $\Sigma_1(L_T)$ they are automatically $\Delta(L_T,T)$
Proof:

$T \vdash \forall \vec{x} \exists! y\ A_j(\vec{x},y)$, so
$T \vdash \neg A_j(\vec{x},y) \leftrightarrow \exists z\ (z \neq y \wedge A_j(\vec{x},z))$

4.8. Lemma
Suppose $T \ll_\Delta U$ ($L_{PA} \subseteq LT$), such that $(\,)^*$ be the translation defined above, then:
$A \in \Delta(L_U,U) \rightarrow A^* \in \Delta(L_T,T)$.
Proof:
i) $c = x$ is trivial.
ii) $y = x$ is also trivial.
iii) $G(t_1,t_2) = x$:

$(G(t_1,t_2) = x)^* \leftrightarrow$

$\exists u_1 \exists u_2 ((t_1 = u_1)^* \wedge (t_2 = u_2)^* \wedge G(u_1,u_2) = x) \longleftrightarrow$ (a)
$\forall u_1 \forall u_2 (((t_1 = u_1)^* \wedge (t_2 = u_2)^*) \rightarrow G(u_1,u_2) = x)$ (b)
(a) is in $\Sigma_1(L_T)$ according to I.H. and
$\neg$ (b) is in $\Sigma_1(L_T)$ according to I.H.

iv) $(F(t) = x)^* \longleftrightarrow$
$\exists u\ ((t = u)^* \wedge A(u,x)) \longleftrightarrow$
$\forall u\ ((t = u)^* \rightarrow A(u,x))$

v) $(t_1 = t_2)^* \longleftrightarrow$
$\exists u\ ((t_1 = u)^* \wedge (t_2 = u)^*) \longleftrightarrow$
$\forall u\ ((t_1 = u)^* \longleftrightarrow (t_2 = u)^*)$

vi) The propositional connectives and "$\exists$" are trivial.

vii) $(\forall x<t\ A(x))^* \longleftrightarrow$
$(\forall x\ (x < t)^* \rightarrow (A(x))^*) \longleftrightarrow$
$(\forall x\ ((\exists u\ (t = u)^* \wedge x < u) \rightarrow (A(x))^*)) \longleftrightarrow$
$\forall u\ ((t = u)^* \rightarrow (\forall x<u\ (A(x))^*)) \longleftrightarrow$ (a)
$\exists u\ ((t = u)^* \wedge \forall x<u\ (A(x))^*)$ (b)
Here $\neg$ (a) $\in \Sigma_1(L_T)$ and (b) $\in \Sigma_1(L_T)$.

4.9. Theorem

$\leq_\Delta$ is a partial order.

4.10. Corollary

Suppose $PA \leq_\Delta T$ and $A \in \Sigma_1(L_T)$ then A^* is provably equivalent in PA with a $\Sigma_1(L_{PA})$ sentence.

Note that $\Delta_0(L_T)$ grows as we go to bigger and bigger Δ-extensions of PA, but $\Delta(L_T,T)$ remains invariant modulo translating back, i.e. $\Delta(L_T,T)^*$ is $\Delta(L_{PA},PA)$ (modulo provable equivalence). Also $\Delta_0(L_T) \subseteq \Delta(L_T,T)$. In fact there is an infinite Δ-extension of PA where $\Delta_0(L_T)$ catches up with $\Delta(L_T,T)$, i.e. each $\Delta(L_T,T)$-formula is provably equivalent in T with a $\Delta_0(L_T)$-formula.

We leave it to the reader to verify that the extensions of PA we defined up till now are indeed Δ-extensions. Sometimes slight adaptations are necessary, e.g.:
$b \subseteq a : \Leftrightarrow \exists c < Sa\ \exists c' < Sa\ a = c * b * c'$

5. Σ_1 - completeness

For propositional calculus, one can effectively decide whether a given formula is derivable or not. When we consider predicate logic (as in PA) this is much more difficult. Just think of a $\Sigma_1(L_{PA})$-sentence $\exists x\ Bx$, where Bx is in $\Delta_0(L_{PA})$. For each r we can check in a finite time whether Br is true or not. But the only procedure one can think of in general to find out if $\exists x\ Bx$ is true, is to check $B_0, B_1, B_2, \cdots$ If it is false we may never know. Now think of a PA substituted for us in the above consideration. Wouldn't it be surprising if PA could prove of every false $\Sigma_1(L_{PA})$-sentence that it was false? Note that we cannot immediately exclude that it can; after all, PA proves of many $\Sigma_1(L_{PA})$-sentences that they are false, e.g. using induction instead of our naive procedure.

5.1. Definition

i) T is complete : $\Leftrightarrow$ For all $A \in L_T$: A is true $\Rightarrow T \vdash A$

ii) T is $\Lambda(L_T)$-complete : $\Leftrightarrow \Lambda \subseteq T$ and
For all $A \in \Lambda$: A is true $\Rightarrow T \vdash A$

In section 8, we will build the Gödel Sentence G. This sentence has the following property:
$\neg$ PA $\vdash G$ and $\neg$ PA $\vdash \neg G$

Accepting no options besides truth and falsity, it follows that PA is incomplete. Worse, we shall see that G is actually true and that $G \in \Pi_1$ thus yielding that PA is $\Pi_1(L_{PA})$-incomplete. It may seem queer at first sight, that PA does not even have the desirable property of completeness. But most of us are fairly familiar with at least the thought of its incompleteness. Who has not heard of Fermat's Last Conjecture? (And secretly tried to solve it.) For all non-zero natural numbers a, b and c: $a^n + b^n \neq c^n$, if $n>2$.

Now we have a surprise. It turns out that PA is $\Sigma_1(L_{PA})$ - complete! Hence, to be incomplete, a subset of PA should be sufficiently rich. (In terms of the hierarchy defined in the previous paragraph.) Before we turn to the proof of this theorem, consider the following important consequence:

5.2. Corollary Let PA $\ll_\Delta T$. Then T is $\Sigma_1(L_T)$-complete.
Proof:

Consider $A \in L_T$: A is true $\Rightarrow A^*$ is true, $\Rightarrow B$ is true, with $B \in \Sigma_1(L_{PA})$ and PA $\vdash A^* \longleftrightarrow B$. (B exists by corollary 4.10) By the $\Sigma_1(L_{PA})$-completeness, we find: PA $\vdash B$ and $PA \vdash (A^* \longleftrightarrow B)$. But then: PA $\vdash A^*$, which yields the desired $T \vdash A$.

5.3. Theorem PA is $\Sigma_1(L_{PA})$-complete.
Proof:

First we prove the theorem for Δ_0-formulas.

In proving the theorem for Δ_0-formulas, two logical connectives $\neg$ and $\rightarrow$ will be bothering us. Fortunately, we can eliminate them as follows: We write $\neg\phi_0 \vee \phi_1$ instead of $\phi_0 \rightarrow \phi_1$, and $\neg$ is pushed inside to the level of atomic formulas. The latter operation is defined by the following functions:
Define $()^+, L_{PA} \rightarrow L_{PA}$,
$()^-, L_{PA} \rightarrow L_{PA}$.

i) $(s = t)^+ := (s = t), (\perp)^- = \perp, (s = t)^- := \neg(s = t), (\perp)^+ = \neg\perp$

ii) $(A \vee B)^+ := (A^+ \vee B^+)$; $(A \wedge B)^+ := (A^+ \wedge B^+)$,
$(A \vee B)^- := (A^- \wedge B^-)$; $(A \wedge B)^- := (A^- \vee B^-)$

$(\neg A)^+ := A^-, (\neg A)^- := A^+$

$(A \rightarrow B)^+ := (A^- \vee B^+), (A \rightarrow B)^- := (A^+ \wedge B^-)$

iii) $(\forall x\, A)^+ := \forall x\, A^+, (\forall x\, A)^- := \exists x\, A^-$

$(\exists x\, A)^+ := \exists x\, A^+, (\exists x\, A)^- := \forall x\, A^-$

Trivially,
PA $\vdash A^+ \longleftrightarrow A$
PA $\vdash A^- \longleftrightarrow \neg A$.

We will prove our theorem for the range of $()^+$ on $\Delta_0(L_{PA})$

Step 1)

$m = n \Rightarrow \mathrm{PA} \vdash \underline{m} = \underline{n}$

Trivial: $\underline{m} = \underline{n}$

Step 2)

$m \neq n \Rightarrow \mathrm{PA} \vdash \underline{m} \neq \underline{n}$

A double Induction.

Step 3)

$m + n = k \Rightarrow \mathrm{PA} \vdash \underline{m} + \underline{n} = \underline{k}$

Induction on n:

- $m + 0 = k \Rightarrow m = k \Rightarrow PA \vdash \underline{m} = \underline{k} \Rightarrow \mathrm{PA} \vdash \underline{m} + \underline{0} = \underline{k}$
- $m + (p + 1) = k \Rightarrow m + p = s$ and $s + 1 = k$
  $\Rightarrow \mathrm{PA} \vdash \underline{m} + \underline{p} = \underline{s}$ and $\mathrm{PA} \vdash \underline{s + 1} = \underline{k}$
  $\Rightarrow \mathrm{PA} \vdash \underline{m} + S\underline{p} = S\underline{s} = \underline{k}$

Step 4)

$m \cdot n = k \Rightarrow \mathrm{PA} \vdash \underline{m} \cdot \underline{n} = \underline{k}$

An exercise for the reader.

Step 5)

$t = k \Rightarrow \mathrm{PA} \vdash t = \underline{k}$

We treat the case $t = t_1 + t_2$ as an example.

$t = k \Rightarrow$ for some m, n $t_1 = m, t_2 = n, m + n = k$
$\Rightarrow \mathrm{PA} \vdash t_1 = \underline{m},$
$\Rightarrow \mathrm{PA} \vdash t_2 = \underline{n},$
$\Rightarrow \mathrm{PA} \vdash \underline{m} + \underline{n} = \underline{k}$
$\Rightarrow \mathrm{PA} \vdash t_1 + t_2 = \underline{k}$

Step 6)

$t_1 = t_2 \Rightarrow \mathrm{PA} \vdash t_1 = t_2$

$t_1 = t_2 \Rightarrow t_1 = k$ and $t_2 = k \Rightarrow \mathrm{PA} \vdash t_1 = \underline{k}$ and $\mathrm{PA} \vdash t_2 = \underline{k}$

Steps 7 through 12 are exercises for the reader.

Step 7)

$t_1 \neq t_2 \Rightarrow \mathrm{PA} \vdash t_1 \neq t_2$

Step 8)

$\bot, \neg \bot$

Step 9,10)

$\vee, \wedge$

Step 11)

$m < n \Rightarrow \mathrm{PA} \vdash \underline{m} < \underline{n}$

Step 12)

$(\forall x < t\ Bx) \Rightarrow PA \vdash \forall x < t\ Bx$

It is sufficient to prove:

$\forall x < p\ Bx \Rightarrow \mathrm{PA} \vdash \forall x < \underline{p}\ Bx$

This is proved by induction on p:

- $p = 0$: trivial.
  (We already proved: $\neg \exists x (x < \underline{0})$
- $p = q + 1$: $\forall x < q + 1\ Bx \Rightarrow (\forall x < q\ Bx)$ and Bq
  $\Rightarrow^{IH} \mathrm{PA} \vdash \forall x < \underline{q}\ Bx$ and $\mathrm{PA} \vdash Bq$
  $\Rightarrow \mathrm{PA} \vdash \forall x < \underline{q}\ Bx \wedge Bq$
  $\Rightarrow \mathrm{PA} \vdash \forall x < \underline{q+1}\ Bx$

This completes the proof for Δ_0 formulas. It is easy to extend the result to $\Sigma_1(L_{PA})$:

$\exists x\ Ax \Rightarrow Ap$, for some p
$\Rightarrow^{IH}$ PA $\vdash Ap$
$\Rightarrow$ PA $\vdash \exists x\ Ax$

6. Primitive recursion

We give a procedure to construct Δ-extensions of PA in which we have symbols for a finite number of desired primitive recursive functions and in which we can prove the defining equations for these functions.

6.1. Definition

The way of defining the beginfunctions is as follows:
0) $A(x_0,\ldots,x_{n-1},y) := (x_i = y)\quad (Pr_i^n)$
$A(x_0,\ldots,x_{n-1},y) := (Sx_i = y)\quad (Sc_i^n)$
$A(x_0,\ldots,x_{n-1},y) := (\underline{m} = y)\quad (Cs_m^n)$

The composition function $Cmp_{n'}^{n}(G,F_0,\ldots,F_{n'-1})$ can be defined as:
1) Suppose $F_0,\ldots,F_{n'-1},G$ are already introduced:
$A(\vec{x},y) := G(F_0(\vec{x}),\ldots,F_{n'-1}(\vec{x})) = y$

The recursion functional $Rec^{n+1}(F,G)$ is defined as :
2) Suppose F,G are already defined:
$A(x_0,\ldots,x_{n-1},y) :=$
$\exists r\ (\mathrm{seq}(r) \wedge \mathrm{length}(r) > x_0 \wedge (r)_0 = F(x_1,\ldots,x_{n-1}) \wedge$
$\forall i{<}\mathrm{length}(r) \dot{-} 1\ ((r)_{i+1} = G((r)_i,i,x_1,\ldots,x_{n-1})) \wedge (r)_{x_0} = y)$

What we have done is that we showed given a Δ-extension of PA, say T where F and G are defined, how to make another Δ-extension, say T' where Rec is defined ($Rec(F,G)(z,\vec{x}) = y \longleftrightarrow A(z,\vec{x},y)$).

Exercise: Check that the definition of Rec is a Σ_1-formula.

6.2. Claim 1

$T \vdash \forall x_0 \exists y\ A(x_0,\vec{x},y)$

Proof: Induction over x_0.

- $r = \underline{0} \circ (\underline{0} \circ F(x_1,\ldots,x_{n-1}))$
  It is obvious that $\mathrm{seq}(r) \wedge (r)_0 = F(x_1,\ldots,x_{n-1}) \wedge \mathrm{length}(r) = \underline{1}$.
- Suppose $A(z,\vec{x},y')$ for a certain y'. Let r' be a sequence as promised by $A(z,\vec{x},y')$.
  If $\mathrm{length}(r') > z{+}1$ then choose $r := r'$.
  Else $r := r' \circ (Sz \circ G(y',z,\vec{x}))$.
  It is easy to see that this works ($y := G(y',z,\vec{x})$).

6.3. Claim 2

$T \vdash A(x_0,\vec{x},y) \wedge A(x_0,\vec{x},y') \rightarrow y = y'$

Proof(in T):

Suppose $A(x_0,\vec{x},y)$ and $A(x_0,\vec{x},y')$ and suppose that r,r' are sequences as promised by these respective formulas, so $(r)_{x_0} = y$, $(r')_{x_0} = y'$. One shows by an easy induction on i: $\forall i < x_0+\underline{1}\ (r)_i = (r')_i$.

6.4. Claim 3

$T \vdash A(0,\vec{x},y) \longleftrightarrow F(\vec{x}) = y$

$T \vdash A(\underline{z+1},\vec{x},y) \longleftrightarrow \exists u\ (A(z,\vec{x},u) \wedge G(u,z,\vec{x}) = y)$

Proof: We do the second half of the claim, the other one is left as an exercise for the reader.

$\leftarrow$ If $\exists u\ (A(z,\vec{x},u) \wedge G(u,z,\vec{x}) = y)$ then the desired sequence is easy to produce from the one promised by $A(z,\vec{x},u)$ and y.

$\rightarrow$ Suppose $A(\underline{z+1},\vec{x},y)$. Consider a sequence r as promised by this formula. Take $u := (r)_z$. Then r is also a witness for $A(z,\vec{x},u)$ such that $y = (r)_{z+1} = G(u,z,\vec{x})$

7. Coding Syntax

In this section, we shall code syntactic objects into numbers. The elementary operations to do this can be expressed by primitive recursion. The process is called Gödel numbering. It will be denoted by $\ulcorner so \urcorner$, where *so* is a syntactic object.

We want to introduce arithmetical operations that mimick on the codes the effect of the syntactical operations on the syntactical objects. For example we construct a primitive recursive 'substitution function' sub_0 that corresponds to the real substitution functions as depicted below:

Term	Variable	Formula		
t	x	ϕ	$\rightarrow$(subst)	$\phi[t/x]$
$\ulcorner t \urcorner$	$\ulcorner x \urcorner$	$\ulcorner \phi \urcorner$	$\rightarrow$(sub0)	$\ulcorner \phi[t/x] \urcorner$

Since concatenation is already defined, we only need to devise an encoding for the characters from PA's alphabet. Apart from the variables, we have finitely many symbols. Although another approach is not expected to lead to disaster, we are going to code variables by writing: x, $'x$, $''x$, $\cdots$, and giving both symbols a code.

The symbols are coded by:

Symbol	(	)	0	x	′	S	+	.	"="
Number	1	2	3	4	5	6	7	8	9
Symbol	$\bot$	$\neg$	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$	$\forall$	$\exists$	
Number	10	11	12	13	14	15	16	17	

For concatenation, we have to choose a base-number (p). Throughout the rest of this section, it will be 17.

In talking about encoded syntactic entities, it would be nice to have a way of identifying the kind of thing a code codes: Is it representing a variable, a term or a formula? In the subsequent definition, predicates characterising these properties are given.

7.1. Definition

a.

$$Var(x) :\Leftrightarrow \exists y < x\ x = var(y),$$

where var is defined to be :

$$var(0) := 4$$
$$var(n+1) := 5 * var(n)$$

b.

$$\begin{aligned} Term(x) :\Leftrightarrow \exists r\ & Seq(r) \wedge \\ & (\forall i < length(r) \\ & [Var((r)_i) \vee \quad (variable) \\ & (r)_i = 3 \vee \quad (zero) \\ & \exists j < i [(r)_i = 6 * (r)_j] \vee \quad (Successor) \\ & \exists j,k < i\ ((r)_i = 1 * (r)_j * 7 * (r)_k * 2) \vee \quad (term_1 + term_2) \\ & \exists j,k < i\ ((r)_i = 1 * (r)_j * 8 * (r)_k * 2) \quad (term_1 \cdot term_2) \\ &] \\ &) \wedge x = (r)_{length(r)-1} \end{aligned}$$

c.

$$\begin{aligned} Form(x) :\Leftrightarrow \exists r\ & Seq(r) \wedge \\ & (\forall i < length(r) \\ & (r)_i = 10 \vee \quad (falsum) \\ & \exists u,v < r\ (Term(u) \wedge Term(v) \wedge (r)_i = 1 * u * 9 * v * 2 \vee \quad (equality) \\ & \exists j < i\ (r)_i = 1 * 11 * (r)_j * 2 \vee \quad (negation) \\ & \exists j,k < i\ \exists z\ 10 < z < 16 \\ & (r)_i = 1 * (r)_j * z * (r)_k * 2 \vee \quad (binary\ connectives) \\ & \exists j < i\ \exists y < r\ \exists z\ 15 < z < 18 \\ & Var(y) \wedge (r)_i = 1 * z * y * (r)_j * 2 \quad (quantifiers) \\ &) \wedge (r)_{length(r)-1} = x \end{aligned}$$

An important property of these definitions is that all formulas are Δ_0. This is the reason that we did not define:

$$Var(x) :\Leftrightarrow \exists y\ x = var(y).$$

As the reader will have recognized, this is a Σ_1-formula.

Now that we know the predicates to recognize a variable, term or formula it is easy to define sub_0, our coding function. Since we want it to be primitively recursive, it has to be total. We therefore use a "don't care" value (zero). In the definition, x is a term, y the variable to be substituted, and z the formula or term in which the substitution should take place.

7.2. Definition

$$sub_0(x,y,z) := \begin{cases} 0 \text{ if } \neg Term(x) \vee \neg Var(y) \vee \neg (Term(z) \vee Form(z)) \\ z \text{ if } Term(x) \wedge Var(y) \wedge (z = 3 \vee (Var(z) \text{and} z \neq y)) \\ x \text{ if } Term(x) \wedge Var(y) \wedge z = y \\ \lceil S \rceil * sub_0(x,y,v) \text{ if } (Term(x) \wedge Var(y) \wedge Term(z) \wedge z = 6 * v \\ etc. \ \cdots \end{cases}$$

The following result is one of may possible results to the effect that our encoding is not ambiguous.

7.3. Theorem: Unique Reading

Define C: $\Leftrightarrow Term(u) \wedge Term(v) \wedge Term(z) \wedge Term(y)$

a. $PA \vdash C \rightarrow \underline{\ulcorner(\urcorner} * u * \underline{\ulcorner+\urcorner} * v * \underline{\ulcorner)\urcorner} \neq \underline{\ulcorner(\urcorner} * z * \underline{\ulcorner\cdot\urcorner} y * \underline{\ulcorner)\urcorner}$

b. $PA \vdash C \wedge (\underline{\ulcorner(\urcorner} * u * \underline{\ulcorner+\urcorner} * v * \underline{\ulcorner)\urcorner} = \underline{\ulcorner(\urcorner} * z * \underline{\ulcorner+\urcorner} * y * \underline{\ulcorner)\urcorner} \rightarrow u = z \wedge v = y$

The phenomenon of Unique Reading may be formulated in many additional results, and the reader is advised to do so for himself. To prove the theorem, a lemma is needed:

7.3.1. Lemma

$t \subseteq_i u \wedge Term(t) \wedge Term(u) \rightarrow t = u$
Proof: induction on t.

proof of unique reading from the lemma

a. trivial.

b. Assume C and $(\underline{\ulcorner(\urcorner} * u * \underline{\ulcorner+\urcorner} * v * \underline{\ulcorner)\urcorner} = \underline{\ulcorner(\urcorner} * z * \underline{\ulcorner+\urcorner} * y * \underline{\ulcorner)\urcorner}$. By result 53, we may cancel the left brackets. We get: $u * \underline{\ulcorner+\urcorner} * v * \underline{\ulcorner)\urcorner} = z * \underline{\ulcorner+\urcorner} * y * \underline{\ulcorner)\urcorner}$, and hence $u \subseteq_i z * \underline{\ulcorner+\urcorner} * y * \underline{\ulcorner)\urcorner}$. Using C and By 54: $u \subseteq_i z$ or $z \subseteq_i u$. applying the lemma: $u = z$. Again from 53 (and associativity) we get $v * \underline{\ulcorner)\urcorner} = y * \underline{\ulcorner)\urcorner}$. Hence by 51: $v = y$.

Recall that we seek to express a self-referential statement. Most ingredients have been prepared by now. We will need to talk about the code of the numeral of a given number. We would naively compute this as follows: given n we first form the numeral of n: $S\,S\,\cdots\,S\underline{0}$ ($nS's$), and then compute the code of this numeral $\ulcorner S\urcorner * \ulcorner S\_\urcorner * \cdots \ulcorner S\urcorner * \ulcorner\underline{0}\urcorner$. Of course we cannot make the intermediate step when working in PA. Happily a direct definition is possible:

$num(0) := \underline{3}$
$num(n+1) := \ulcorner S\urcorner * num(n)$

Note that this is only a sketch; the definition is not in the formal language either. However, a formulation using primitive recursion is easily found.

To facilitate our reasoning, define another substitution function:

$sub(x,y) := sub_0(num(x),4,y)$

All free occurrences of x in y are replaced by the syntactic entity num(x).

Example:

$sub(10, \ulcorner(x + x) = 'x\urcorner) = \ulcorner\underline{10} + \underline{10} = 'x\urcorner$

7.4. Fixed point theorem

(Gödel)

Let $A(x,\vec{y})$ be a formula. There is a formula $B(\vec{y})$, such that: PA $\vdash A(\ulcorner\underline{B}\urcorner,\vec{y}) \longleftrightarrow B(\vec{y})$

proof:
Define $C := A(sub(x, x))$

$B := A(sub(\underline{\ulcorner C \urcorner}, \underline{\ulcorner C \urcorner}))$

PA $\vdash sub(\underline{\ulcorner C \urcorner}, \underline{\ulcorner C \urcorner}) =^{(defC)}$

$sub(\underline{\ulcorner C \urcorner}, \underline{\ulcorner A(sub(x,x)) \urcorner}) =^{(defsub)}$

$\ulcorner A(sub(\underline{\ulcorner C \urcorner}, \underline{\ulcorner C \urcorner})) \urcorner =^{(defB)} \underline{\ulcorner B \urcorner}$

Ergo, PA $\vdash A(sub(\underline{\ulcorner C \urcorner}, \underline{\ulcorner C \urcorner})) \leftrightarrow A(\underline{\ulcorner B \urcorner})$

8. Predicates for Proof-codes

Now that we have encoded all syntactic objects, we wish to represent the notion of provability in PA-predicates. Since it seems to be quite difficult to code natural deduction-trees, we shall use sequents. To simplify a bit, we use a restricted language without $\exists$ and $\vee$. This does not restrict our results! Both can be expressed in the remaining connectives.

8.1. Definition

Sequents are of the form $\Gamma \vdash \phi$, where Γ is a finite set of L_{PA}-formulas, and ϕ is a L_{PA}-formula. The set of sequents is the smallest set such that:

i)

a) $\Gamma \cup \{\phi\} \vdash \phi$ is a sequent

b) $\varnothing \vdash \phi$, with ϕ an axiom of PA is a sequent

ii)

a) Let $\Gamma_1 \vdash \phi$, $\Gamma_2 \vdash \psi$ be sequents, then $\Gamma_1 \cup \Gamma_2 \vdash \phi \wedge \psi$ is a sequent.

b) Let $\Gamma \vdash \phi \wedge \psi$ be a sequent. Then $\Gamma \vdash \phi$ and $\Gamma \vdash \psi$ are sequents.

c) Let $\Gamma \cup \{\phi\} \vdash \psi$ be sequent, then $\Gamma \vdash \phi \rightarrow \psi$ is a sequent.

d) Let $\Gamma_1 \vdash \phi$ and $\Gamma_2 \vdash \phi \rightarrow \psi$ be sequents, then $\Gamma_1 \cup \Gamma_2 \vdash \psi$ is a sequent.

e) Let $\Gamma \vdash \bot$ be a sequent, then $\Gamma \vdash \phi$ is a sequent.

f) Let $\Gamma \cup \{\neg\phi\} \vdash \bot$ be a sequent, then $\Gamma \vdash \phi$ is a sequent.

g) Let $\Gamma \vdash \forall x\, \phi(x)$ be a sequent. Then $\Gamma \vdash \phi(t)$ is a sequent, provided that t is free for x in ϕ.

h) Let $\Gamma \vdash \phi(x)$, x not occuring free in any $\psi \in \Gamma$, be a sequent, then $\Gamma \vdash \forall x \phi(x)$ is a sequent.

This provides a good starting point to introduce predicates for notions related to proofs like Sequent, Axiom and Rule. These predicates are in the language of PA. They take numerals as arguments and return true if the corresponding numbers code the desired notion.

8.2. Definition

$Sequent(x) :\Leftrightarrow Seq(x) \wedge Length(x) = 2 \wedge$
$\forall i < Length((x)_0)(Form((x)_0)_i) \wedge$
$Form((x)_1)$

$Axiom(x) :\Leftrightarrow Sequent(x) \wedge$

$$(\exists i < Length((x)_0)((x_0))_i = (x)_1) \vee$$
$$(PA-axiom((x)_1)))$$

$$Rule(x,y,z): \Leftrightarrow Sequent(x) \wedge Sequent(y) \wedge Sequent(z)$$
$$(\underline{\wedge I}(x,y,z) \vee$$
$$\underline{\wedge E}(x,y,z) \vee$$
$$\underline{\rightarrow I}(x,y,z) \vee$$
$$\underline{\rightarrow E}(x,y,z) \vee$$
$$\underline{\perp}(x,y,z) \vee$$
$$\underline{RAA}(x,y,z) \vee$$
$$\underline{\forall I}(x,y,z) \vee$$
$$\underline{\forall E}(x,y,z))$$

As an example, we give the definition of $\underline{\wedge I}(x,y,z)$. The reader should try some others form himself.

$$\underline{\wedge I}(x,y,z): \Leftrightarrow (z)_1 = \underline{\ulcorner(\urcorner} * (x)_1 * \underline{\ulcorner\wedge\urcorner} * (y)_1 * \underline{\ulcorner)\urcorner} \wedge$$
$$(\forall i < Length((x)_0)\ \exists j < Length((z)_0)((x)_0)_i = ((z)_0)_j)$$
$$(\forall i < Length((y)_0)\ \exists j < Length((z)_0)((y)_0)_i = ((z)_0)_j)$$

0 is representing the empty set of assumptions. The definition is tailored to be of Δ_0 formulas forms. Now we come to defining predicates for proof and provability.

8.3. Definition

i.)

$$Proof_0(r,z): \Leftrightarrow Seq(r) \wedge$$
$$(\forall i < Length(r)$$
$$Axiom((r)_i) \vee$$
$$\exists j,k < i\ Rule((r)_j,(r)_k,(r)_i) \wedge$$
$$r_{Length(r)\dot{-}1} = z)$$

("r is a proof of z", r an encoded set of sequents, z an encoded sequent)

ii.)

$$Proof(r,x): \Leftrightarrow \exists z < r\ Proof_0(r,z) \wedge$$
$$(z)_0 = \underline{0} \wedge$$
$$(z)_1 = x$$

("r is a proof of x", r an encoded set of sequents, x an encoded formula)

iii.)

$$Prov(x): \Leftrightarrow \exists y\ Proof(y,x)$$

("x is provable in PA", x an encoded predicate in the language of PA)

The first and second clause in this definition still yield Δ_0-formulas. The third is a Σ_1 formula.

To streamline our notation still further, we introduce the following abbreviation:

$$\Box\phi: \Leftrightarrow Prov(\underline{\ulcorner\phi\urcorner})$$

We can talk about provability in PA in PA itself! The fruits are ready to be picked:

8.4. Corollary
There is a sentence G such that:
PA $\vdash G \longleftrightarrow \neg \Box G$
(This sentence is the famous Gödel Sentence)
Proof:
Apply the Fixed Point Lemma to $\neg\, Prov(x)$. We get: There is a G with PA $\vdash G \longleftrightarrow \neg \Box G$

Note that G is (provably equivalent to) a Π_1-sentence.

8.5. Theorem (Löb)

i) PA $\vdash \phi \Rightarrow$ PA $\vdash \Box\phi$

ii) PA $\vdash \Box\phi \rightarrow \Box\Box\phi$

iii) PA $\vdash \Box(\phi \rightarrow \psi) \rightarrow (\Box\phi \rightarrow \Box\psi)$

Proof:

i) Assume PA $\vdash \phi$, then $\Box\phi$ is true. We know that $\Box\phi$ is Σ_1. Hence, by Σ_1 completeness: PA $\vdash \Box\phi$

ii) By a formalisation of the proof of Σ_1-completeness. For $A \in \Sigma_1$: PA $\vdash A \rightarrow \Box A$. We will not carry out the details here.

iii) Trivial.

Now, after so many pages, so much dreary encoding, we have reached the end of our journey. We are ready to prove the incompleteness theorems.

8.6. First Incompleteness Theorem
$PA \vdash G \Rightarrow PA \vdash \bot$
$PA \vdash \neg G \Rightarrow PA \vdash \Box\bot$
Proof:

$$PA \vdash G \Rightarrow \left\{\begin{matrix} PA \vdash \Box G \ (Th.\ 8.5\ ) \\ PA \vdash \neg\Box G\,(Cor.\ 8.4\ ) \end{matrix}\right\} \Rightarrow PA \vdash \bot$$

$$PA \vdash \neg G \Rightarrow \left\{\begin{matrix} PA \vdash \Box G\,(Cor.8.4) \\ PA \vdash \Box\neg G\,(Th.8.5) \end{matrix}\right\} \Rightarrow PA \vdash \Box\bot\ (Th.\ 8.5\ )$$

8.7.
Second Incompleteness Theorem
$PA \vdash \neg\Box\bot \Rightarrow PA \vdash \bot$
Proof:
We show that: $PA \vdash G \longleftrightarrow \neg\Box\bot$.

$PA \vdash \Box\bot \rightarrow \Box G$
$(PA \vdash \bot \rightarrow G$
$PA \vdash \Box(\bot \rightarrow G)(Th.\ 8.5\ .i)$
$PA \vdash (\Box\bot \rightarrow \Box G)(Th.\ 8.5\ .iii))$

Hence, $PA \vdash G \rightarrow \neg \Box G \rightarrow \neg \Box \bot$.

Now, we show that: $PA \vdash \Box G \rightarrow \Box \bot$

$$PA \vdash \Box G \rightarrow \left\{ \begin{matrix} \Box\Box G \\ \Box\neg \Box G \end{matrix} \right\} \rightarrow \Box \bot$$

($PA \vdash G \rightarrow \neg \Box G$ (*Cor.* 8.4)
$PA \vdash \Box(G \rightarrow \neg \Box G)$ (*Th.* 8.5.*i*)
$PA \vdash (\Box G \rightarrow \Box\neg \Box G)$ (*Th.* 8.5.*iii*))

Ergo: $PA \vdash \neg \Box \bot \rightarrow \neg \Box G \rightarrow G$

LITERATURE

Reference:

[1] Alex S. Jeffrey: Computability and Logic. G.S. Boolos and R.C. Jeffrey. Cambridge University Press, Cambridge, 1980.

Not quoted in the text.

1. Kurt Gödel, Collected Works, Volume I, Publications 1929–1936. S. Feferman et. al. (eds.) Oxford University Press, Oxford, 1986.

2. Self reference and modal logic. C. Smoryński. Springer Verlag, New York, 1985.