

GÖDEL'S INCOMPLETENESS THEOREMS 2018/2019

MARCUS TRESSL

Lecture Notes

<http://personalpages.manchester.ac.uk/staff/Marcus.Tressl/teaching/Goedel/index.php>

CONTENTS

1.	<u>Recursive functions</u>	1
1.1.	Definition and the Church-Turing thesis	1
1.2.	The pairing function and Gödel's β -function	7
1.3.	Primitive recursion	11
1.4.	Sequence numbers	15
1.5.	Recursively enumerable sets	22
2.	<u>Formal proofs and the completeness theorem</u>	24
2.1.	Languages and formulas	24
2.2.	Structures and Tarski's definition of truth	31
2.3.	Logical axioms and the definition of a formal proof	33
2.4.	Soundness and the Completeness Theorem	35
2.5.	Propositional Tautologies and the Prenex Normal Form	36
3.	<u>Representation of recursive functions in arithmetic</u>	39
4.	<u>Arithmetisation of Logic: Gödelisation</u>	47
5.	<u>Undecidability and incompleteness</u>	52
5.1.	Recursively axiomatizable and decidable theories	52
5.2.	The first incompleteness theorem	56
5.3.	Undecidable sentences	60
6.	<u>Applications to decision problems</u>	64
6.1.	Interpretations	65
6.2.	Strongly undecidable structures	68
7.	<u>The Arithmetic Hierarchy</u>	72
7.1.	The structure of arithmetic formulas	72
7.2.	Recursion revisited	75
7.3.	Kleene's Enumeration theorem	77
7.4.	Hilbert's 10 th problem	78
8.	<u>Gödel's second incompleteness theorem</u>	82
	References	85
	Index	86

1. RECURSIVE FUNCTIONS

1.1. Definition and the Church-Turing thesis.

We first need to formalize the intuitive notion of a program or an algorithm. At the moment we think of a program as a device that receives finite tuples of natural numbers as input and returns a natural number as output, according to some rules. This will be done implicitly, i.e. we will define what a program or an algorithm is supposed to accomplish and then we may think of a program as a machine who can carry out this task.

The set-up of input and output data being natural numbers seems rather restrictive. However we will later introduce a coding process which will explain why our special setup covers all conceivable programs.

- The set of natural numbers including 0 is denoted by ω , whereas \mathbb{N} denotes $\{1, 2, 3, \dots\}$. For $n \in \omega$, the set ω^n denotes the set of all n -tuples of elements of ω ; thus for $n = 0$, ω^n is just $\{\emptyset\}$.
- For any $n \in \omega$ and any subset R of ω^n we also use the notation $R(a)$ for $a \in R$ and think of R as an n -ary relation of ω . Further, we write $\mathbb{1}_R$ for the characteristic function of R in ω^n , i.e. the function $\mathbb{1}_R \rightarrow \{0, 1\}$ with $\mathbb{1}_R^{-1}(1) = R$, hence

$$\mathbb{1}_R(a) = 1 \iff a \in R \iff R(a).$$

- For $1 \leq i \leq n \in \mathbb{N}$, let $I_i^n : \omega^n \rightarrow \omega$ be the projection onto the i^{th} coordinate, hence

$$I_i^n(a_1, \dots, a_n) = a_i.$$

The I_i^n are also called **coordinate functions**.

- For any **non-empty** subset R of ω let

$$\mu x(R) := \min(R)$$

denote the smallest element of R . If we think of R as a property of natural numbers including 0 it is convenient to write

$$\mu x(R(x)) \text{ instead of } \mu x(\{n \in \omega \mid n \text{ has property } R\}).$$

We now define a class of functions of which we think as being computable by a program, i.e. for each of these functions f , there is a program that accepts inputs a from the domain of f and returns $f(a)$ as output.

1.1.1. Definition. A function $\omega^m \rightarrow \omega$ (where $m \in \mathbb{N}$) is called **recursive** or **computable**, if it is obtained by a finite iteration from the following rules:

R1 Elementary functions: Each of the following functions is recursive.

- for all $i \leq n$, all coordinate functions I_i^n .
- addition and multiplication of ω , both being functions $\omega^2 \rightarrow \omega$.
- the characteristic function $\mathbb{1}_{\leq} : \omega^2 \rightarrow \omega$ of the order relation \leq of ω .

R2 Composition rule:

If $n, k \in \mathbb{N}$ and $F : \omega^n \rightarrow \omega$ and $G_1, \dots, G_n : \omega^k \rightarrow \omega$ are recursive functions, then also the composition

$$F \circ (G_1, \dots, G_n) : \omega^k \rightarrow \omega$$

is a recursive function. We will also write $F(G_1, \dots, G_n)$ for this composition. Thus $F(G_1, \dots, G_n)$ maps $a = (a_1, \dots, a_k)$ to $F(G_1(a), \dots, G_n(a))$.

R3 μ -recursion (or minimalisation or search):

If $n \in \mathbb{N}$ and $F : \omega^n \times \omega \rightarrow \omega$ is a recursive function such that for every $a \in \omega^n$ there is some $x \in \omega$ with $F(a, x) = 0$, then also the function $G : \omega^n \rightarrow \omega$ defined by

$$G(a) := \mu x (F(a, x) = 0) \quad (\text{recall, this is } \min\{x \in \omega \mid F(a, x) = 0\}).$$

is recursive.

A set (or a relation) $R \subseteq \omega^n$ is called **recursive**, if its characteristic function $\mathbb{1}_R : \omega^n \rightarrow \omega$ is recursive.

It is very plausible that every particular recursive function $F : \omega^n \rightarrow \omega$ can be computed by some machine, e.g.: On any modern computer (ignoring time and space limitations) one can in principle write a program that accepts input a from ω^n and returns $F(a)$. Surprisingly we have the following


Church-Turing thesis:

Every function $\omega^n \rightarrow \omega$ that can be computed *somehow*, is recursive.

This is a philosophical statement and the term "can be computed somehow" is informal. It comprises all possible ways a function $F : \omega^n \rightarrow \omega$ could be manufactured by some device. We will first develop some basic properties of recursive functions and come back with more evidence for the Church-Turing thesis later on.

In case you have seen Turing machines before: They all manufacture recursive functions and conversely, every recursive function can be computed with a Turing machine. In this sense the notions recursive and computable coincide. We will not talk about concrete implementations here (as we do not need it) and omit this interesting part of the theory.

Note that the content of our course does not depend on whether the Church-Turing thesis holds true, i.e. we do not depend on it in any way.

 Please read definition 1.1.1 carefully. Firstly, observe that only functions $\mathbb{1}_S$ defined on *all* of ω^m are recursive in this course. Hence if $S \subseteq \omega^m$ is a proper subset and $f : S \rightarrow \omega$ is a function, then it does not make sense to say that f is recursive. Obviously some of such functions are computable (in the naive sense): For example set $m = 2$, $S = \{(a, b) \in \omega^2 \mid a \geq b\}$; then the subtraction, considered as a function $f : S \rightarrow \omega$ (hence f is defined by $f(a, b) = a - b$) is computable in the naive sense. However, this function is not recursive in the sense of the

definition 1.1.1 of recursivity. The reason why we exclude such 'partially defined' functions in the first place has merely technical reasons: It makes the theory easier. Computability (in the naive sense) of such 'partially defined' functions is captured in this course by saying that the graph of f (which a priori is a subset of $S \times \omega$) is a recursive subset of $\omega^m \times \omega$.

A second warning is related to the clause "by a finite iteration" at the beginning of 1.1.1. The objects that are defined in 1.1.1 are functions $\omega^m \rightarrow \omega$. And these objects have to be produced in finitely many steps. Hence, for example the function 2^n is a priori not obtained from a *finite* application of the composition rule **R2** starting with multiplication (given in **R1**). Make sure you understand this reasoning. Still, after some work we will see in 1.2.5 that the function $f : \omega \rightarrow \omega$, $f(n) = 2^n$ is recursive.

In this subsection we will look at some basic constructions and show that they give new recursive functions.

1.1.2. Lemma.

- (i) Let $m, n \in \mathbb{N}$, let $F : \omega^n \rightarrow \omega$ be recursive and let $\sigma : \{1, \dots, n\} \rightarrow \{1, \dots, m\}$ be a map. Then also the map $F^\sigma : \omega^m \rightarrow \omega$ defined by

$$F^\sigma(x_1, \dots, x_m) := F(x_{\sigma(1)}, \dots, x_{\sigma(n)})$$

is recursive.

Thus, for example, the map defined by

$$(x_1, \dots, x_{n+1}) \mapsto F(x_{n+1}, \underbrace{x_1, x_1, \dots, x_1}_{n-1 \text{ entries}})$$

is recursive.

- (ii) The relations \geq and $=$ of ω^2 are recursive.
 (iii) If $F_1, \dots, F_k : \omega^n \rightarrow \omega$ and $R \subseteq \omega^k$ are recursive, then also

$$R(F_1, \dots, F_k) := \{a \in \omega^n \mid R(F_1(a), \dots, F_k(a))\}$$

(which is the preimage of R under the map $(F_1, \dots, F_k) : \omega^n \rightarrow \omega^k$) is recursive.

- (iv) For all $i \in \omega$, the constant function

$$c_i^n : \omega^n \rightarrow \omega$$

with value i is recursive.

Proof. (i) follows from **R2**, because

$$F^\sigma = F(I_{\sigma(1)}^m, \dots, I_{\sigma(n)}^m).$$

(ii) follows from **R1** and (i) because $\mathbb{1}_{\geq}(x, y) = \mathbb{1}_{\leq}(y, x)$ and $\mathbb{1}_=(x, y) = \mathbb{1}_{\leq}(x, y) \cdot \mathbb{1}_{\geq}(x, y)$.

(iii) follows from **R2** because

$$\mathbb{1}_{R(F)} = \mathbb{1}_R(F_1, \dots, F_k).$$

(iv). We have

- $c_1^1 = \mathbb{1}_{\leq}(I_1^1, I_1^1)$ and $c_0^1 = \mathbb{1}_{\leq}(I_1^1 + c_1^1, I_1^1)$.
- $c_{i+1}^1 = c_i^1 + c_1^1$ and $c_i^n = c_i^1(I_1^n)$.

□

1.1.3. Corollary. *Let $F : \omega^n \rightarrow \omega$ be recursive.*

- (i) *F considered as a function $\omega^n \times \omega^m \rightarrow \omega$ is recursive. Formally: The function $G : \omega^n \times \omega^m \rightarrow \omega$ defined by $G(a, b) = F(a)$ is recursive.*
- (ii) *For $m \leq n$ and $a \in \omega^m$, also the function $F(a, x_1, \dots, x_{n-m}) : \omega^{n-m} \rightarrow \omega$ is recursive.*
- (iii) *If $G : \omega^n \rightarrow \omega$ is another recursive function then the sets given by the n -ary relations $F(x) = G(x)$, $F(x) \leq G(x)$ and $F(x) \geq G(x)$ of ω are recursive.*

Proof. (i) follows from 1.1.2(i) because $G = F^\sigma$ for the inclusion $\sigma : \{1, \dots, n\} \hookrightarrow \{1, \dots, n + m\}$. Item (ii) follows from 1.1.2(iv) and **R2**. Item (iii) follows from 1.1.2(ii),(iii). \square

1.1.4. Boolean operations

For $R, S \subseteq \omega^n$ we write

$$\begin{aligned} \neg R &:= \omega^n \setminus R, \quad R \wedge S := R \cap S, \quad R \vee S := R \cup S \text{ and} \\ R \rightarrow S &:= (\neg R) \vee S, \quad R \leftrightarrow S := (R \rightarrow S) \wedge (S \rightarrow R). \end{aligned}$$

If R and S are recursive, then all these relations are recursive.

Proof. If R and S are recursive, then $R \wedge S$ is recursive, since $\mathbb{1}_{R \wedge S} = \mathbb{1}_R \cdot \mathbb{1}_S$. The relation $\neg R$ is recursive, since $\neg R(a) \iff \mathbb{1}_R(a) = c_0^n(a)$ (now use 1.1.3(iii)).

All the other relations defined here are built up from conjunction and negation, hence they are recursive, too. \square

1.1.5. Examples.

- (i) The binary relations $<$ and $>$ on ω are recursive, because these are the complements of \geq and \leq respectively (use 1.1.4 and also see 1.1.2).
- (ii) For every $a = (a_1, \dots, a_n) \in \omega^n$ the singleton subset $\{a\}$ is recursive as $x_1 = a_1 \wedge \dots \wedge x_n = a_n$ is recursive¹. Consequently, by 1.1.4, all finite and all cofinite² subsets of ω^n are recursive.

1.1.6. Definitions by cases

Let $R_1, \dots, R_k \subseteq \omega^n$ be recursive such that for every $a \in \omega^n$ there is a unique $i \in \{1, \dots, k\}$ with $R_i(a)$.

- (i) If $F_1, \dots, F_k : \omega^n \rightarrow \omega$ are recursive, then also the function $F : \omega^n \rightarrow \omega$ defined by

$$F(a) = \begin{cases} F_1(a) & \text{if } R_1(a) \\ \vdots & \\ F_k(a) & \text{if } R_k(a) \end{cases}$$

is recursive.

- (ii) If $S_1, \dots, S_k \subseteq \omega^n$ are recursive, then also the relation $S \subseteq \omega^n$ defined by

$$S(a) \iff \begin{cases} S_1(a) & \text{if } R_1(a) \\ \vdots & \\ S_k(a) & \text{if } R_k(a) \end{cases}$$

¹Hence $\{a\}$ is the intersection of n recursive subsets of ω^n . Make sure you understand this argument

²A subset S of a set X is called **cofinite** if $X \setminus S$ is finite

is recursive.

Proof. (i) follows from **R2** and

$$F = F_1 \cdot \mathbb{1}_{R_1} + \dots + F_k \cdot \mathbb{1}_{R_k}.$$

(ii) follows from 1.1.4 and

$$S = (S_1 \wedge R_1) \vee \dots \vee (S_k \wedge R_k).$$

□

Now we start using μ -recursion.

1.1.7. μ -recursion for relations

Let $R \subseteq \omega^n \times \omega$ such that for every $x \in \omega^n$ there is some $y \in \omega$ with $R(x, y)$. Thus we may define a function $F : \omega^n \rightarrow \omega$ by

$$F(x) = \min\{y \mid R(x, y)\}.$$

Then

- (i) If R is recursive, then F is recursive.
- (ii) If R is the graph of a function, then F is this function and R is recursive if and only if F is recursive.

Proof. (i) follows from μ -recursion because

$$F(x) = \mu y (\mathbb{1}_{\neg R}(x, y) = 0)$$

and because we know that with R also $\neg R$ is recursive (see 1.1.4).

(ii) By (i) it remains to show that the graph of F is recursive if F is a recursive function. This follows from 1.1.3(iii)

$$R(a, b) \iff \tilde{F}(a, b) = I_{n+1}^{n+1}(a, b),$$

where $\tilde{F}(a, b) := F(a)$ (which is also recursive by 1.1.3(i)).

□

We shall see later on that 1.1.7 can indeed not be proved without μ -recursion.

We also need subtraction, whenever defined:

1.1.8. Almost subtraction

The map $\dot{-} : \omega^2 \rightarrow \omega$ defined by

$$a \dot{-} b := \begin{cases} a - b & \text{if } a \geq b \\ 0 & \text{if } a < b. \end{cases}$$

is recursive.

Proof. We have

$$a \dot{-} b = \mu x (b + x = a \vee a < b).$$

Observe that the ternary relation $R(a, b, x)$ defined by $b + x = a \vee a < b$ is recursive and for all $a, b \in \omega$, there is some $x \in \omega$ with $R(a, b, x)$. Hence we may apply μ -recursion for relations, 1.1.7. □

1.1.9. Bounded μ -recursion

For $S \subseteq \omega$ and $b \in \omega$ we define

$$\mu_{x < b}(S) \text{ or } \mu_{x < b}(S(x))$$

to be the least $x \in S$ with $x < b$ if there is such an x and b otherwise. The letter ‘ b ’ here indicates a bound. So

$$\mu_{x < b}(S) = \min(S \cup \{b\}), \text{ also written as } \mu_{x < b}(S) = \mu x(S(x) \vee x = b)$$

For $R \subseteq \omega^n \times \omega$ we define $F_R : \omega^n \times \omega \rightarrow \omega$ by

$$F_R(a, b) := \mu_{x < b}(R(a, x)).$$

Since $F_R(a, b) := \mu x(R(a, x) \vee x = b)$, F_R is recursive if R is recursive (by μ -recursion for relations, 1.1.7).

For better readability we shall write $\mu_{x \leq b}$ instead of $\mu_{x < b+1}$.

1.1.10. Bounded quantification

Let $R \subseteq \omega^n \times \omega$ be recursive. The relations $E_R, A_R \subseteq \omega^n \times \omega$ defined by

$$E_R(a, b) \iff \exists x < b R(a, x) \text{ and}$$

$$A_R(a, b) \iff \forall x < b R(a, x)$$

are recursive.

Proof. Define F_R as in bounded μ -recursion, 1.1.9. Then

$$E_R(a, b) \iff F_R(a, b) < b \text{ and } A_R(a, b) \iff F_{\neg R}(a, b) = b.$$

□

Of course we shall also write $\exists x \leq b$ instead of $\exists x < b + 1$ and $\forall x \leq b$ instead of $\forall x < b + 1$.

1.1.11. **Lemma.** *The ternary relation*

$$n \equiv m \pmod{d}$$

(defined by $n - m \in d \cdot \mathbb{Z}$) is recursive.

Proof. We have $n \equiv m \pmod{d}$ if and only if

$$\exists x \leq n (n = x \cdot d + m) \quad \vee \quad \exists x \leq m (m = x \cdot d + n).$$

Hence from 1.1.10 (and 1.1.4 for the disjunction) we get the lemma. □

1.2. The pairing function and Gödel's β -function.

1.2.1. **Theorem and Definition.** *The Pairing Function* $\text{Pair} : \omega^2 \longrightarrow \omega$, defined by

$$\text{Pair}(x, y) := \frac{(x + y)(x + y + 1)}{2} + x$$

is recursive and bijective. The compositional inverse of Pair is denoted by

$$(L, R) : \omega \longrightarrow \omega^2.$$

The functions L and R are recursive and given as follows: For $n \in \omega$, let $z \leq n$ be maximal with $\frac{z(z+1)}{2} \leq n$. Then

$$L(n) = n - \frac{z(z+1)}{2} \text{ and } R(n) = z + \frac{z(z+1)}{2} - n.$$

Further, we have the following properties of Pair , L and R :

- (i) For all $x, y, n \in \omega$ we have $x, y \leq \text{Pair}(x, y)$ and $L(n), R(n) \leq n$.
- (ii) $\text{Pair}(0, 0) = L(0) = R(0) = 0$.
- (iii) If $n \neq 0$, then $L(n) < n$.

Proof. Take $n \in \omega$ and let $z \in \omega$ be maximal with $\frac{z(z+1)}{2} \leq n$. Then $n < \frac{(z+1)(z+2)}{2}$ and therefore

$$0 \leq n - \frac{z(z+1)}{2} < \frac{(z+1)(z+2)}{2} - \frac{z(z+1)}{2}.$$

Since $\frac{(z+1)(z+2)}{2} - \frac{z(z+1)}{2} = z + 1$ we have

$$\begin{aligned} 0 \leq x := n - \frac{z(z+1)}{2} &\leq z \text{ and so} \\ y := z - x &\geq 0 \end{aligned}$$

Now

$$\text{Pair}(x, y) = \frac{(x + y)(x + y + 1)}{2} + x = \frac{z(z+1)}{2} + x = n,$$

as required.

Pair is injective:

If also $\text{Pair}(x', y') = n$, then $\frac{(x'+y')(x'+y'+1)}{2} \leq n$ and so by choice of z , $x'+y' \leq z$. If $x'+y' < z$, then

$$\begin{aligned} n = \text{Pair}(x', y') &= \frac{(x' + y')(x' + y' + 1)}{2} + x' \leq \frac{(z-1)z}{2} + z - 1 < \\ &< \frac{z(z+1)}{2} \leq \text{Pair}(x, y) = n, \text{ a contradiction.} \end{aligned}$$

Thus $x' + y' = z$, which implies $x' = n - \frac{(x'+y')(x'+y'+1)}{2} = n - \frac{(x+y)(x+y+1)}{2} = x$ and then $y' = (x' + y') - x' = (x + y) - x = y$.

This shows that Pair is bijective. Pair is recursive because addition, multiplication and the function $f : \omega \longrightarrow \omega$,

$$n \mapsto \begin{cases} \frac{n}{2} & \text{if } n \text{ is even,} \\ \frac{n-1}{2} & \text{if } n \text{ is odd} \end{cases}$$

are recursive: we have

$$f(n) = \mu_{\leq n} x (2x > n) \dot{-} 1.$$

In the proof of surjectivity, the definitions of x and y show that $L(n) = x$ and $R(n) = y$ can be written as claimed. Properties (i)-(iii) are readily verified and they imply that L and R are recursive using bounded μ -recursion 1.1.9 and bounded quantification 1.1.10:

$$L(n) = \mu_{\leq n} x (\exists y \leq n \text{ Pair}(x, y) = n) \text{ and}$$

$$R(n) = \mu_{\leq n} y (\exists x \leq n \text{ Pair}(x, y) = n).$$

□

We need a fundamental fact from elementary number theory, we include a proof for the sake of completeness.

1.2.2. Chinese remainder theorem

Let $k_1, \dots, k_n \in \mathbb{Z}$ be pairwise coprime (i.e., for all $i \neq j$, no natural number > 1 divides k_i and k_j). Then, for all $a_1, \dots, a_n \in \mathbb{Z}$, there is some $x \in \omega$, $x \leq k_1 \cdot \dots \cdot k_n$ that solves all congruences

$$x \equiv a_1 \pmod{k_1}$$

$$\vdots$$

$$x \equiv a_n \pmod{k_n}.$$

Proof. First note that for a solution x in \mathbb{Z} of all the congruences in our theorem, we may add or subtract a suitable multiple of $k_1 \cdot \dots \cdot k_n$ to x and obtain another solution which is in the range $\{0, 1, 2, \dots, k_1 \cdot \dots \cdot k_n\}$. Hence it suffices to find a solution $x \in \mathbb{Z}$.

We do this by induction on n starting from $n = 1$, where we may choose $x = a_1$. Now assume we know the theorem for $n - 1$. Hence there are $y, \gamma_1, \dots, \gamma_{n-1} \in \mathbb{Z}$ with

$$(\dagger) \quad y = a_i + \gamma_i \cdot k_i \quad (1 \leq i < n).$$

As k_n is coprime to each of k_1, \dots, k_{n-1} , it is also coprime to $k_1 \cdot \dots \cdot k_{n-1}$. Using the euclidean algorithm, we therefore can find $\alpha, \beta \in \mathbb{Z}$ with $1 = \alpha k_n + \beta k_1 \dots k_{n-1}$. Multiplying this through by $y - a_n$, there are also $\alpha, \beta \in \mathbb{Z}$ with

$$y - a_n = \alpha k_n + \beta k_1 \dots k_{n-1}.$$

We take

$$x = y - \beta k_1 \dots k_{n-1} = a_n + \alpha k_n$$

and see that $x = a_n + \alpha k_n \equiv a_n \pmod{k_n}$. But from (\dagger) we also see for all $i < n$ that

$$x = y - \beta k_1 \dots k_{n-1} = a_i + \gamma_i \cdot k_i - \beta k_1 \dots k_{n-1} \equiv a_i \pmod{k_i}.$$

□

1.2.3. Lemma. (Gödel)

Let $\beta^* : \omega^3 \rightarrow \omega$ be defined by

$$\beta^*(a, b, i) := \mu x \left(x \equiv a \pmod{1 + b(i+1)} \right).$$

So $\beta^*(a, b, i)$ is the remainder when we divide a by $1 + (i+1)b$.

Then β^* is recursive and for each $n \in \omega$ and all $a_0, \dots, a_{n-1} \in \omega$ there is some $(a, b) \in \omega^2$ with

$$\beta^*(a, b, i) = a_i \quad (0 \leq i < n).$$

In fact we can choose $b = n! \cdot (1 + a_0 + \dots + a_{n-1})$ and find such an a with $a \leq (1 + nb)^n$.

Proof. β^* is recursive by 1.1.11 and μ -recursion.

Pick $b \in \omega$, $b \geq a_0, \dots, a_{n-1}$, such that $p|b$ for all primes $p \leq n$. For example $b = n! \cdot (1 + a_0 + \dots + a_{n-1})$ will do. We first show that $1 + b, 1 + 2b, \dots, 1 + nb$ are pairwise coprime.

Let $1 \leq i < j \leq n$ and suppose p is a prime that divides $1 + ib$ and $1 + jb$. Then $p|(j - i)b$, so $p|j - i$ or $p|b$. As $j - i \leq n$, the choice of b implies $p|b$ in either case. But then p cannot be a divisor of $1 + ib$,

By the Chinese Remainder Theorem, there is some $a \in \omega$ with $a \leq (1 + nb)^n$ such that

$$\begin{aligned} a &\equiv a_0 \pmod{1 + b} \\ a &\equiv a_1 \pmod{1 + 2b} \\ &\dots \\ a &\equiv a_{n-1} \pmod{1 + nb} \end{aligned}$$

From the choice of b we have $a_i \leq b < 1 + (i + 1)b$. But then a_i is the smallest $x \in \omega$ with

$$a \equiv x \pmod{1 + (i + 1)b}.$$

By definition of β^* we see that $\beta^*(a, b, i) = a_i$ ($0 \leq i < n$). \square

1.2.4. Gödel's β -function

We define Gödel's β -function $\beta : \omega^2 \rightarrow \omega$ by

$$\beta(a, i) := \beta^*(L(a), R(a), i).$$

β has the following properties:

- (1) β is recursive.
- (2) $\beta(a, i) \leq a + 1$ for all $a, i \in \omega$.
- (3) For each $n \in \omega$ and all $a_0, \dots, a_{n-1} \in \omega$ there is some $a \in \omega$ with

$$\beta(a, 0) = a_0, \beta(a, 1) = a_1, \dots, \beta(a, n - 1) = a_{n-1}.$$

Proof. (1) By 1.1.11, β^* is recursive. By 1.2.1 also L and R are recursive. Hence β is a composition of recursive function and therefore recursive itself.

(2) For $a = 0$ we have $L(a), R(a) = 0$ and $\beta(a, i) = \beta^*(0, 0, i) = 0$.

Now let $a > 0$. Since $\beta^*(a, b, i) < 1 + (i + 1)b$ by definition and $\beta^*(a, b, i)$ is congruent to $a \pmod{1 + (i + 1)b}$ it is clear that

$$\beta^*(a, b, i) \leq a$$

for all $a, b, i \in \omega$. Since $L(a) < a$ for $a > 0$ (see 1.2.1(iii)) we get

$$\beta(a, i) = \beta^*(L(a), R(a), i) \leq L(a) < a.$$

(3) is immediate from the corresponding property of β^* in 1.2.3. \square

It should also be noted that in the literature, sometimes the function β^* from 1.2.3 is called Gödel's β -function.

So β 'codes' finite sequences. It will also allow us to code finite sequences of formulas (in an appropriate language) and to attach natural numbers to 'formal'

proofs (yet to be defined). In the next two sections we will see that β gives us a strong tool to produce new recursive functions. An ad hoc example right away:

1.2.5. *Examples.* The function $\omega \rightarrow \omega$, $n \mapsto 2^n$ is recursive (please also reread the second warning before 1.1.2) To see this, we use the function $g : \omega \rightarrow \omega$ defined by

$$g(n) := \mu x \left(\beta(x, 0) = 1 \wedge \forall i < n \beta(x, i + 1) = 2\beta(x, i) \right).$$

Hence $g(n)$ is the smallest natural number x that defines a sequence a_0, \dots, a_n via β , that satisfies $a_0 = 1$ and $a_{i+1} = 2a_i$ ($i < n$). g is recursive, as β is recursive, bounded quantification is recursive (see 1.1.10) and μ -recursion is applicable (i.e. there is indeed such a natural number). Since

$$2^n = \beta(g(n), n),$$

our claim is proved. The same reasoning also shows that the function $n \mapsto n!$ is recursive. Just modify g to

$$g(n) := \mu x \left(\beta(x, 0) = 1 \wedge \forall i < n \beta(x, i + 1) = (i + 1)\beta(x, i) \right).$$

1.2.6. *Remark.* Observe that the only properties we needed in the examples 2^n and $n!$ above, were properties (1)-(3) in 1.2.4. This will also be the case for the rest of this course when it comes to the generation of recursive functions (see also 1.3.9).

Another function that has properties (1)-(3) of 1.2.4 is the function $V : \omega^2 \rightarrow \omega$ defined by

$V(a, n) =$ the largest $d \in \omega$ such that p_{n+1}^d divides a , where p_n is the n^{th} prime .
(In algebraic terms: $V(a, n)$ is the p_{n+1} -adic valuation of a). This function is somewhat easier to understand than our choice of β , so why did we not choose this function as β ? The reason is that we need to show that V is recursive and this proof already refers to a β -function. The proof that V is recursive will be done in question 9 of the example sheet.

1.3. Primitive recursion.

We will single out a subset of recursive functions now which play an important role in computability theory, but also in theoretical questions on how to represent recursive functions in the natural numbers; more on that topic will follow in subsequent sections.

1.3.1. Definition. For $n \in \omega$ let $h : \omega^n \rightarrow \omega$ and $H : \omega^n \times \omega \times \omega \rightarrow \omega$ be functions. We say that a function $F : \omega^n \times \omega \rightarrow \omega$ is **obtained by primitive recursion from H with initial value h** if the following two conditions hold:

- (i) For all $a \in \omega^n$ we have $F(a, 0) = h(a)$.
- (ii) For all $(a, b) \in \omega^n \times \omega$ we have $F(a, b + 1) = H(a, b, F(a, b))$.

Hence $H(a, b, i)$ should be considered as the “recursion-rule” for F . It computes $F(a, b + 1)$ from assuming $F(a, b)$ is known.

To understand the roles of the parameters a, b in this definition look at the following examples.

1.3.2. Examples. The function $b \mapsto b!$ and exponentiation $(a, b) \mapsto a^b$ are primitive recursive. The map $b!$ is obtained by primitive recursion with initial value 1 from $H(b, x) = (b + 1) \cdot x$ (so here $n = 0$ in 1.3.1). Exponentiation (for $a > 0$) is obtained by primitive recursion with initial value 1 from $H(a, b, x) = a \cdot x$ (here $n = 1$ in 1.3.1).

1.3.3. Theorem. Let $h : \omega^n \rightarrow \omega$ and $H : \omega^n \times \omega \times \omega \rightarrow \omega$ be recursive functions and let $F : \omega^n \times \omega \rightarrow \omega$ be obtained by primitive recursion from H with initial value h . Then also F is recursive.

Proof. To see this, we use the function $G : \omega^n \times \omega \rightarrow \omega$ defined by

$$G(a, b) := \mu x \left(\beta(x, 0) = h(a) \wedge \forall i < b \beta(x, i + 1) = H(a, i, \beta(x, i)) \right).$$

Hence $G(a, b)$ is the smallest natural number x that defines a sequence $k_0, \dots, k_b \in \omega$ via β , that satisfies $k_0 = h(a)$ and $k_{i+1} = H(a, i, k_i)$ for all $i < b$. Clearly, in this sequence we have $F(a, 0) = h(a) = k_0$ and by induction $F(a, i + 1) = H(a, i, F(a, i)) = H(a, i, k_i) = k_{i+1} = \beta(x, i + 1)$.

The function G is recursive, as β is recursive, bounded quantification is recursive (see 1.1.10) and μ -recursion is applicable (i.e. there is indeed such a natural number x as described).

But now we see that F is recursive as well, because $F(a, b) = \beta(G(a, b), b)$. \square

1.3.4. Definition. A function $\omega^m \rightarrow \omega$ is called **primitive recursive** if it is obtained by a finite iteration from the rules **R1**, **R2** of 1.1.1 and from the rule

PR If $F : \omega^n \times \omega \rightarrow \omega$ is obtained by primitive recursion from $H : \omega^n \times \omega \times \omega \rightarrow \omega$ with initial value $h : \omega^n \rightarrow \omega$ (see 1.3.1), and if h and H are primitive recursive, then also F is primitive recursive.

A set (or a relation) $R \subseteq \omega^n$ is called **primitive recursive**, if its characteristic function $\mathbb{1}_R : \omega^n \rightarrow \omega$ is primitive recursive.

1.3.5. Observation.

The statements 1.1.2 – 1.1.6 are also true if we replace recursive by primitive recursive everywhere. This is by inspection of the proof of these statements: We have only used **R1** and **R2** in these proofs.

1.3.6. Lemma. Let $F : \omega^n \times \omega \rightarrow \omega$ be a function.

(i) We write ΣF and ΠF for the functions $\omega^n \times \omega \rightarrow \omega$ defined by

$$(\Sigma F)(a, b) = \sum_{i < b} F(a, i) \text{ and}$$

$$(\Pi F)(a, b) = \prod_{i < b} F(a, i).$$

If F is (primitive) recursive, then also ΣF and ΠF are (primitive) recursive.

(ii) We write $\text{Sup}F$ and $\text{Inf}F$ for the functions $\omega^n \times \omega \rightarrow \omega$ defined by

$$(\text{Sup}F)(a, b) = \sup\{F(a, i) \mid i \leq b\} \text{ and}$$

$$(\text{Inf}F)(a, b) = \inf\{F(a, i) \mid i \leq b\}.$$

If F is (primitive) recursive, then also $\text{Sup}F$ and $\text{Inf}F$ are (primitive) recursive.

Proof.

(i) ΣF is obtained by primitive recursion from $H(a, b, x) = F(a, b) + x$ with initial value c_0^n :

$$(\Sigma F)(a, b + 1) = F(a, b) + (\Sigma F)(a, b) = H(a, b, (\Sigma F)(a, b)).$$

ΠF is obtained by primitive recursion from $H(a, b, x) = F(a, b) \cdot x$ with initial value c_1^n :

$$(\Pi F)(a, b + 1) = F(a, b) \cdot (\Pi F)(a, b) = H(a, b, (\Pi F)(a, b)).$$

(ii) $\text{Sup}F$ is obtained by primitive recursion with initial value $F(a, 0)$ from

$$H(a, b, x) = \begin{cases} F(a, b + 1) & \text{if } F(a, b + 1) \geq x, \\ x & \text{otherwise.} \end{cases}$$

Note that Definition by Cases 1.1.6 is primitive recursive. Similarly, $\text{Inf}F$ is obtained by primitive recursion with initial value $F(a, 0)$ from

$$H(a, b, x) = \begin{cases} F(a, b + 1) & \text{if } F(a, b + 1) \leq x, \\ x & \text{otherwise.} \end{cases}$$

□

1.3.7. Proposition. Bounded quantification (1.1.10), bounded μ -recursion (1.1.9) and almost subtraction (1.1.8) are also true if we replace recursive by primitive recursive everywhere.

Proof. Proof of bounded μ -recursion for primitive recursive functions:

Let $R \subseteq \omega^n \times \omega$ be primitive recursive. We need to show that $F_R : \omega^n \times \omega \rightarrow \omega$ defined by

$$F_R(a, b) := \mu x_{< b}(R(a, x)),$$

is primitive recursive. This is so, because $\neg R$ is primitive recursive and

$$F_R(a, b) = \sum_{i < b} \prod_{j \leq i} \mathbb{1}_{\neg R}(a, j) = (\Sigma G)(a, b),$$

where $G(a, i) = (\prod \mathbb{1}_{\neg R})(a, i + 1)$. Hence by 1.3.6(i), also F_R is primitive recursive.

Proof of almost subtraction for primitive recursive functions:

We already know that bounded μ -recursion is primitive recursive. Hence

$$a \dot{-} b = \mu x_{\leq a} (b + x = a \vee a < b)$$

is primitive recursive.

Proof of bounded quantification for primitive recursive functions:

Take $R \subseteq \omega^n \times \omega$ primitive recursive. Recall that the relations $E_R, A_R \subseteq \omega^n \times \omega$ are defined by

$$\begin{aligned} E_R(a, b) &\iff \exists x < b \ R(a, x) \text{ and} \\ A_R(a, b) &\iff \forall x < b \ R(a, x). \end{aligned}$$

We have

$$\begin{aligned} \mathbb{1}_{E_R}(a, b) &= \begin{cases} (\text{Sup} \mathbb{1}_R)(a, b \dot{-} 1) & \text{if } 0 < b \\ 0 & \text{if } b = 0. \end{cases} \quad \text{and} \\ \mathbb{1}_{A_R}(a, b) &= \begin{cases} (\text{Inf} \mathbb{1}_R)(a, b \dot{-} 1) & \text{if } 0 < b \\ 1 & \text{if } b = 0. \end{cases} \end{aligned}$$

Hence by 1.3.6(ii), also E_R and A_R are primitive recursive \square

1.3.8. Proposition.

- (i) The pairing function, the components L and R of its compositional inverse and the functions β and β^* (see 1.2.4 and 1.2.3) are primitive recursive.
- (ii) Further, there is a primitive recursive function $\rho : \omega \rightarrow \omega$ such that for all $N, n \in \omega$ and all $a_0, \dots, a_{n-1} \in \omega$ with $n, a_0, \dots, a_{n-1} \leq N$, there is some $x \in \omega$ with $x < \rho(N)$ such that $\beta(x, 0) = a_0, \dots, \beta(x, n-1) = a_{n-1}$.

Proof. We only need to assemble what we already know about these functions:

(i) The ternary relation

$$n \equiv m \pmod{d}$$

is primitive recursive, because it is defined by bounded quantification, see 1.1.11.

Since

$$\beta^*(a, b, i) = \mu x_{< 1+b(i+1)} \left(x \equiv a \pmod{1+b(i+1)} \right)$$

β^* is primitive recursive. The pairing function is obviously primitive recursive. Then, the proof of the recursiveness of L and R in 1.2.1 uses a definition of these functions in terms of the pairing function, bounded quantification and bounded μ -recursion. Hence L and R are primitive recursive. Since $\beta(x, i) = \beta^*(L(x), R(x), i)$, also β is primitive recursive.

(ii) We define

$$\rho(N) = \text{Pair}((N+2)!, (N+2)!^N \cdot N^N),$$

which is primitive recursive. Take $N, n, a_0, \dots, a_{n-1} \in \omega$ with $n, a_0, \dots, a_{n-1} \leq N$. By 1.2.3 we know that there is some $x \in \omega$ with $L(x) = n! \cdot (1 + a_0 + \dots + a_{n-1})$ and $R(x) \leq (1 + nn! \cdot (1 + a_0 + \dots + a_{n-1}))^n$ such that $\beta(x, 0) = a_0, \dots, \beta(x, n-1) = a_{n-1}$.

Then

$$n! \cdot (1 + a_0 + \dots + a_{n-1}) \leq (n+1)! \cdot N \leq (N+2)! \text{ and}$$

$$(1 + nn! \cdot (1 + a_0 + \dots + a_{n-1}))^n \leq (n+2)!^n N^n \leq (N+2)!^N \cdot N^N.$$

Now

$$x = \text{Pair}(L(x), R(x)) \leq \text{Pair}((N+2)!, (N+2)!^N \cdot N^N) = \rho(N)$$

as required. \square

1.3.9. *Remark.* As mentioned in 1.2.6, only properties (1)-(3) in 1.2.4 of β are needed to manufacture recursive functions in this course. On the other hand, if we want to check that certain functions are primitive recursive, we use our particular β -function. For example, 1.3.8(ii) will be used when we want to apply bounded μ -recursion in connection with the β -function in order to verify that a certain function is primitive recursive. We shall see an example in the next section (1.4.2(iii)).

1.4. Sequence numbers.

1.4.1. **Definition.** For $n \in \omega$ and $(a_1, \dots, a_n) \in \omega^n$ we define the **sequence number** of (a_1, \dots, a_n) as

$$\prec a_1, \dots, a_n \succ := \mu x \left(\beta(x, 0) = n \wedge \forall i < n \beta(x, i+1) = a_{i+1} \right).$$

By 1.2.4(3) this indeed makes sense. So $\prec a_1, \dots, a_n \succ$ is the smallest natural number x such that

$$(n, a_1, \dots, a_n) = (\beta(x, 0), \beta(x, 1), \dots, \beta(x, n)).$$

We extend this definition by setting the sequence number of the empty sequence to $\prec \succ := 0$.

Using 1.3.8(ii) and the primitive recursive function ρ used there we see that in fact

$$\prec a_1, \dots, a_n \succ := \mu x_{\leq \rho(1+n+a_1+\dots+a_n)} \left(\beta(x, 0) = n \wedge \forall i \leq n \beta(x, i) = a_i \right).$$

Since bounded μ -recursion and bounded quantification are primitive recursive, the function $\omega^n \rightarrow \omega$ defined by

$$(a_1, \dots, a_n) \mapsto \prec a_1, \dots, a_n \succ$$

is primitive recursive.

1.4.2. Operations with sequence numbers

(i) **The length function ℓ .**

The function $\ell : \omega \rightarrow \omega$ defined by $\ell(a) = \beta(a, 0)$ is primitive recursive and is called the **length function**. $\ell(a)$ is called the **length** of a . It should be stressed that the function ℓ can be applied to all $a \in \omega$, but it only expresses something meaningful when a is a sequence number $\prec a_1, \dots, a_n \succ$; in that case

$$\ell(\prec a_1, \dots, a_n \succ) = n.$$

If $a \in \omega$ is not a sequence number, $\ell(a)$ has no meaning for us. However, in recursive definitions later on, which use the ℓ -function, it is important that ℓ is *defined* for all $a \in \omega$. A similar remark also applies to all the other construction in 1.4.2.

(ii) **The coordinate function.**

The function $\omega^2 \rightarrow \omega$ that maps (a, i) to $\beta(a, i)$ is called the i -th coordinate function. It is obviously primitive recursive and written as

$$(a)_i := \beta(a, i).$$

Since $\beta(a, i) \leq a \pm 1$ we have

$$\begin{aligned} (\prec a_1, \dots, a_n \succ)_i &= a_i < \prec a_1, \dots, a_n \succ \text{ for } 1 \leq i \leq n \text{ and} \\ (\prec a_0, \dots, a_{n-1} \succ)_{i+1} &= a_i < \prec a_0, \dots, a_{n-1} \succ \text{ for } i < n. \end{aligned}$$

Observe that $(a)_0 = \ell(a)$.

(iii) **The concatenation function.**

The function $\omega^2 \rightarrow \omega$, defined by

$$a \hat{\ } b = \mu x_{\leq \rho(a+b+1)} \left(\begin{aligned} \ell(x) &= \ell(a) + \ell(b) \wedge \\ \forall 1 \leq i \leq \ell(a) \ (x)_i &= (a)_i \wedge \\ \forall 1 \leq j \leq \ell(b) \ (x)_{\ell(a)+j} &= (b)_j \end{aligned} \right)$$

is called the concatenation function. We make use here of the primitive recursive function $\rho : \omega \rightarrow \omega$ from 1.3.8(ii): Obviously, there is some $x \in \omega$ that has the property in the bounded μ -operator above; since $\ell(a) + \ell(b)$, $(a)_i$, $(b)_i \leq a + b$ for all $a, b, i \in \omega$, we can find such an $x \leq \rho(a + b + 1)$ (by 1.3.8(ii)).

Hence $a \hat{\ } b$ is defined by bounded μ -recursion using primitive recursive functions, and so it is primitive recursive itself. By definition we have:

$$\langle a_1, \dots, a_n \rangle \hat{\ } \langle b_1, \dots, b_k \rangle = \langle a_1, \dots, a_n, b_1, \dots, b_k \rangle$$

for all $k, n, a_1, \dots, a_n, b_1, \dots, b_k \in \omega$.

(iv) **The restriction function.**

The function $\omega^2 \rightarrow \omega$ that maps (a, i) to $\langle \beta(a, 1), \dots, \beta(a, i) \rangle$ is called the i -th restriction function and written as

$$a \upharpoonright_i := \langle \beta(a, 1), \dots, \beta(a, i) \rangle .$$

Hence for $i \leq n$ we have

$$\langle a_1, \dots, a_n \rangle \upharpoonright_i = \langle a_1, \dots, a_i \rangle .$$

This function is primitive recursive because it is obtained by primitive recursion with initial value $h(a) = 0$ (which is the sequence number of the empty tuple) from

$$H(a, i, x) = x \hat{\ } \langle \beta(a, i + 1) \rangle ,$$

which is primitive recursive by (iii). We have

$$a \upharpoonright_{i+1} := \langle \beta(a, 1), \dots, \beta(a, i + 1) \rangle = a \upharpoonright_i \hat{\ } \langle \beta(a, i + 1) \rangle = H(a, i, a \upharpoonright_i).$$

(v) Finally we define

$$\mathbf{Seq} := \{x \in \omega \mid x \text{ is a sequence number}\}$$

and verify that \mathbf{Seq} is a primitive recursive subset of ω : We have

$$\mathbf{Seq}(a) \iff \forall x < a \ \exists i \leq \ell(a) \ (x)_i \neq (a)_i.$$

(Recall that $(x)_0$ is the length of x)

1.4.3. Definition. For every map $F : \omega^n \times \omega \rightarrow \omega$ let $\bar{F} : \omega^n \times \omega \rightarrow \omega$ be defined by

$$\bar{F}(a, b) := \langle F(a, 0), \dots, F(a, b - 1) \rangle .$$

In particular, $\bar{F}(a, 0) = \langle \rangle = 0$. So \bar{F} satisfies

$$(\bar{F}(a, b))_{i+1} = F(a, i) \text{ for all } i < b.$$

1.4.4. Proposition. *A map $F : \omega^n \times \omega \rightarrow \omega$ is (primitive) recursive if and only if \bar{F} is (primitive) recursive.*

Proof. If \bar{F} is (primitive) recursive, then F is (primitive) recursive, because

$$F(a, b) = (\bar{F}(a, b + 1))_{b+1}.$$

and the coordinate functions are primitive recursive.

Conversely, if F is (primitive) recursive, then \bar{F} is (primitive) recursive, because \bar{F} is obtained by primitive recursion from $h(a) = 0$ and $H(a, b, x) = x \hat{\prec} F(a, b) \succ$:

$$\begin{aligned} \bar{F}(a, b + 1) &= \prec F(a, 0), \dots, F(a, b) \succ = \\ &= \prec F(a, 0), \dots, F(a, b - 1) \succ \hat{\prec} F(a, b) \succ = H(a, b, \bar{F}(a, b)). \end{aligned}$$

Recall that concatenation $x \hat{\prec} y$ is a primitive recursive function $\omega^2 \rightarrow \omega$. □

Our main tool to produce recursive functions later on is given by the following.

1.4.5. Recursion on previous values

Let $G : \omega^n \times \omega \times \omega \rightarrow \omega$ be a function. Then there is a unique function $F : \omega^n \times \omega \rightarrow \omega$ with

$$F(a, b) = G(a, b, \bar{F}(a, b)) \quad ((a, b) \in \omega^n \times \omega).$$

If G is (primitive) recursive then also F is (primitive) recursive.

Proof. Existence and uniqueness of F follow by induction from

$$\begin{aligned} F(a, 0) &= G(a, 0, 0) \\ F(a, b + 1) &= G(a, b + 1, \prec F(a, 0), \dots, F(a, b) \succ). \end{aligned}$$

Now assume G is (primitive) recursive. Then \bar{F} is obtained by primitive recursion with initial value $h(a) = 0$ from $H(a, b, x) = x \hat{\prec} G(a, b, x) \succ$, because

$$\begin{aligned} \bar{F}(a, b + 1) &= \prec F(a, 0), \dots, F(a, b) \succ = \bar{F}(a, b) \hat{\prec} F(a, b) \succ = \\ &= \bar{F}(a, b) \hat{\prec} G(a, b, \bar{F}(a, b)) \succ = H(a, b, \bar{F}(a, b)). \end{aligned}$$

Hence \bar{F} is (primitive) recursive and so by 1.4.4, also F is (primitive) recursive. □

1.4.6. *Example.* Let $D_1 : \omega^n \times \omega \times \omega \rightarrow \omega$ and $E, D_2 : \omega^n \times \omega \rightarrow \omega$ be (primitive) recursive. Clearly there is a unique function $F : \omega^n \times \omega \rightarrow \omega$ with

$$F(a, b) = \begin{cases} D_1(a, b, F(a, D_2(a, b))) & \text{if } D_2(a, b) < b \\ E(a, b) & \text{otherwise.} \end{cases}$$

and F is (primitive) recursive.

Proof. Define

$$G(a, b, x) = \begin{cases} D_1(a, b, (x)_{D_2(a, b)+1}) & \text{if } D_2(a, b) < b, \\ E(a, b) & \text{otherwise.} \end{cases}$$

By recursion on previous values, there is a (primitive) recursive function $F : \omega^n \times \omega \rightarrow \omega$ with

$$F(a, b) = G(a, b, \bar{F}(a, b)) = \begin{cases} D_1(a, b, (\bar{F}(a, b))_{D_2(a, b)+1}) & \text{if } D_2(a, b) < b, \\ E(a, b) & \text{otherwise.} \end{cases}$$

As $(\bar{F}(a, b))_{D_2(a, b)+1} = F(a, D_2(a, b))$ we see that this function F is the function we have defined in this example. □

So it turns out that all constructions we have done so far, except μ -recursion itself, are primitive recursive, and the question arises whether it is possible to obtain μ -recursion from primitive recursion. In the final part of this section we will show that this is not the case, so there is a recursive function that is not primitive recursive. This function also has other surprising properties and is useful for recursion theory; however we shall not use this function later on and the rest of section 1.4 below is not examinable.

1.4.7. The Ackermann function

We define functions $A_n : \omega \rightarrow \omega$ by induction on n as follows. Let

$$\begin{aligned} A_0(y) &= y + 1 \\ A_{n+1}(0) &= A_n(1) \text{ and inductively} \\ A_{n+1}(y+1) &= A_n(A_{n+1}(y)) \end{aligned}$$

Let $A(n, y) := A_n(y)$. Then $A : \omega^2 \rightarrow \omega$ is called the **Ackermann function**.

For example we have

- $A(0, y) = y + 1$
- $A(1, y) = y + 2 = 2 + (y + 3) - 3$
- $A(2, y) = 2y + 3 = 2(y + 3) - 3$
- $A(3, y) = 2^{y+3} - 3$
- \vdots
- $A(4, y) = 2^{2^{\cdot^{\cdot^{\cdot}}}} - 3$ ($y + 3$ terms)
- $A(5, y)$ cannot be expressed in everyday notation. Using Donald Knuth's up-arrow notation for large numbers we can write $A(4, y) = 2 \uparrow\uparrow (y + 3) - 3$, $A(5, y) = 2 \uparrow\uparrow\uparrow (y + 3) - 3$, $A(6, y) = 2 \uparrow\uparrow\uparrow\uparrow (y + 3) - 3$, etc.

1.4.8. Proposition.

(i) If $F : \omega^n \rightarrow \omega$ is primitive recursive, then there is some $N \in \omega$ such that

$$F(x_1, \dots, x_n) \leq A_N(x_1 + \dots + x_n) \text{ for all } x_1, \dots, x_n \in \omega$$

(ii) The function $A(x, x)$ is eventually larger than any primitive recursive function $F : \omega \rightarrow \omega$, hence there is some $n \in \omega$ such that $A(x, x) > F(x)$ for all $x > n$. In particular, neither $A(x, y)$ nor $A(x, x)$ are primitive recursive.

(iii) The graph of A is primitive recursive, hence $A(x, y)$ and $A(x, x)$ are recursive.

We will prove this after some preparation.

1.4.9. Lemma. *The function A is strictly increasing in each variable and for all n and x, y we have:*

- (i) $A_n(x + y) \geq A_n(x) + y$;
- (ii) $n \geq 1 \Rightarrow A_{n+1}(y) > A_n(y) + y$;
- (iii) $A_{n+1}(y) \geq A_n(y + 1)$;
- (iv) $2A_n(y) < A_{n+2}(y)$;
- (v) $x < y \Rightarrow A_n(x + y) \leq A_{n+2}(y)$.

Proof. Assume inductively that A_0, \dots, A_n are strictly increasing and that $A_0(y) < A_1(y) < \dots < A_n(y)$ for all y . Then

$$A_{n+1}(y + 1) = A_n(A_{n+1}(y)) \geq A_0(A_{n+1}(y)) > A_{n+1}(y),$$

so A_{n+1} is strictly increasing. Next we show that $A_{n+1}(y) > A_n(y)$ for all y : $A_{n+1}(0) = A_n(1)$, so $A_{n+1}(0) > A_n(0)$ and $A_{n+1}(0) > 1$, so $A_{n+1}(y) > y + 1$ for all y . Hence $A_{n+1}(y + 1) = A_n(A_{n+1}(y)) > A_n(y + 1)$.

Inequality (i) follows easily by induction on n , and a second induction on y .

For inequality (ii), we proceed again by induction on (n, y) : Using $A_1(y) = y + 2$ and $A_2(y) = 2y + 3$, we obtain $A_2(y) > A_1(y) + y$. Let $n > 1$, and assume inductively that $A_n(y) > A_{n-1}(y) + y$. Then $A_{n+1}(0) = A_n(1) > A_n(0) + 0$, and

$$\begin{aligned} A_{n+1}(y + 1) &= A_n(A_{n+1}(y)) \geq A_n(y + 1 + A_n(y)) \geq A_n(y + 1) + A_n(y) > \\ &> A_n(y + 1) + y + 1. \end{aligned}$$

In (iii) we proceed by induction on y . We have equality for $y = 0$. Assuming inductively that (iii) holds for a certain y we obtain

$$A_{n+1}(y + 1) = A_n(A_{n+1}(y)) \geq A_n(A_n(y + 1)) \geq A_n(y + 2).$$

Note that (iv) holds for $n = 0$. For $n > 0$ we have by (i), (ii) and (iii):

$$A_n(y) + A_n(y) \leq A_n(y + A_n(y)) < A_n(A_{n+1}(y)) = A_{n+1}(y + 1) \cdot A_{n+2}(y).$$

Note that (v) holds for $n = 0$. Assume (v) holds for a certain n . Let $x < y + 1$. We can assume inductively that if $x < y$, then $A_{n+1}(x + y) \leq A_{n+3}(y)$, and we want to show that

$$A_{n+1}(x + y + 1) \leq A_{n+3}(y + 1).$$

Case 1. $x = y$. Then

$$\begin{aligned} A_{n+1}(x + y + 1) &= A_{n+1}(2x + 1) = A_n(A_{n+1}(2x)) \leq A_{n+2}(2x) < \\ &< A_{n+2}(A_{n+3}(x)) = A_{n+3}(y + 1). \end{aligned}$$

Case 2. $x < y$. Then

$$A_{n+1}(x + y + 1) = A_n(A_{n+1}(x + y)) \leq A_{n+2}(A_{n+3}(y)) = A_{n+3}(y + 1).$$

□

1.4.10. Proof of 1.4.8(i)

We write $|x|$ instead of $x_1 + \dots + x_n$ if $x = (x_1, \dots, x_n)$.

1.4.8(i) is clear for the successor function $y + 1$, any coordinate function and for any characteristic function. Addition is obviously obtained with primitive recursion from the successor function and multiplication is obtained with primitive recursion from addition. It therefore suffices to see that the claim is preserved when we apply **R2** and primitive recursion **PR**.

So assume we have $N \in \omega$ and functions F, G_1, \dots, G_n with

$$F(x_1, \dots, x_n) \leq A_N(|x|) \text{ and } G_i(y_1, \dots, y_k) \leq A_N(|y|)$$

globally. Then

$$\begin{aligned} F(G_1(|y|), \dots, G_n(|y|)) &\leq A_N(G_1(|y|) + \dots + G_n(|y|)) \leq \\ &\leq A_N(2^n \cdot A_N(|y|)) \leq \text{by 1.4.9}(iv) \\ &\leq A_N(\cdot A_{N+2n}(|y|)) \leq \\ &\leq A_{N+2n+1}(|y|). \end{aligned}$$

Now assume that F is obtained by primitive recursion from $H(x, y, z)$ with initial value $h(x)$. Suppose $N \in \omega$ so that $h(x) \leq A_{N+3}(|x|)$ $H(x, y, z) \leq A_N(|x| + |y| + z)$. Then A_{N+3} also bounds F :

$$\begin{aligned} F(x, y + 1) &= H(x, y, F(x, y)) \leq A_N(|x| + y + A_{N+3}(|x| + y)) \leq \text{by 1.4.9}(v) \\ &\leq A_{N+2}(A_{N+3}(|x| + y)) = A_{N+3}(|x| + y + 1). \end{aligned}$$

□

So we know item (i) of 1.4.8. Item (ii) of 1.4.8 then follows easily and $A(x, y)$ is not primitive recursive. However $A(x, y)$ is recursive, and indeed the graph of A is primitive recursive. To see this, first note that A is obtained by a double recursion:

$$\begin{aligned} A(0, y) &= y + 1, \quad A(x + 1, 0) = A(x, 1) \\ A(x + 1, y + 1) &= A(x, A(x + 1, y)). \end{aligned}$$

1.4.11. Proof of 1.4.8(iii)

Let us call a sequence $a = (a_0, \dots, a_{n-1}) \in (\omega \times \omega \times \omega)^n$ an Ackermann calculation, if the following conditions hold true:

- (1) If $(0, y, z)$ is an entry of a (witnessing $A(0, y) = z$), then $z = y + 1$
- (2) If $(x + 1, 0, z)$ is an entry of a (witnessing $A(x + 1, 0) = z$), then also $(x, 1, z)$ is an entry of a (witnessing $A(x, 1) = z$).
- (3) If $(x + 1, y + 1, z)$ is an entry of a (witnessing $A(x + 1, y + 1) = z$), then there is an entry (x', y', z') of a such that
 - $x' = x + 1$ and $y' = y$ (witnessing $A(x + 1, y) = z'$) and
 - (x, z', z) is an entry of a (witnessing $A(x, z') = z$ and therefore also witnessing $A(x + 1, y + 1) = z = A(x, z') = A(x, A(x + 1, y))$)

1.4.12. **Lemma.** *The following are equivalent for every triple $(x, y, z) \in \omega^3$.*

- (i) $A(x, y) = z$
- (ii) (x, y, z) is an entry of an Ackermann calculation
- (iii) There is an Ackermann calculation $(a_1, \dots, a_{(z+1)^3})$ such that all entries of all a_i are $\leq z$ and (x, y, z) is one of the a_i .

Proof. Let $\Gamma \subseteq \omega^3$ be the graph of the Ackermann function.

(i) \Rightarrow (iii). Let $a = (a_1, \dots, a_{(z+1)^3})$ be an enumeration (possibly with repetition) of the set

$$\Gamma \cap \{0, \dots, z\}^3.$$

We show that a is an Ackermann calculation and (x, y, z) is one of the a_i . The latter statement follows from $A(x, y) = z$ and the fact that $x, y \leq A(x, y) = z$. Now we need to check that conditions (1)-(3) for an Ackermann calculation hold true.

- (1) If $(0, v, w)$ is an entry of a , then $A(0, v) = w$ and so $w = v + 1$ as required.
- (2) If $(u + 1, 0, w)$ is an entry of a , then $A(u + 1, 0) = w$, so by definition of A , $w = A(u, 1)$. Since $(u, 1) \leq w \leq z$ we see that also $(u, 1, w)$ is an entry of a .
- (3) If $(u + 1, v + 1, w)$ is an entry of a , then $w = A(u, A(u + 1, v))$. Since A is monotone in the second coordinate we see that

$$w' := A(u + 1, v) \leq A(u + 1, v + 1) = w \leq z.$$

Hence $(u + 1, v, w')$ is an entry of a and with $A(u, w') = w$, also (u, w', w) is an entry of A as required.

(iii) \Rightarrow (ii) is trivial.

(ii) \Rightarrow (i). We show by induction on x that for every (x, y, z) that appears in an Ackermann calculation a we have $A(x, y) = z$.

If $x = 0$, then by condition (1) of an Ackermann calculation we know that $z = y + 1$. Hence $z = A(0, y)$.

$x \rightarrow x + 1$. So we assume that we know (ii) \Rightarrow (i) for all (u, v, w) with $u \leq x$ and we show (ii) \Rightarrow (i) for $(x + 1, y, z)$ by induction on y .

$y = 0$: Assume $(x + 1, y, z)$ is an entry of a . By condition (2) of an Ackermann calculation we know that $(x, 1, z)$ is an entry of a . Hence by induction on x we see that $z = A(x, 1) = A(x + 1, 0)$ as required.

$y \rightarrow y + 1$. Assume $(x + 1, y + 1, z)$ is an entry of a . By condition (3) of an Ackermann calculation, there is some $z' \in \omega$ such that $(x + 1, y, w)$ and (x, z', z) are entries of a . By induction on y this means $z' = A(x + 1, y)$ and by induction on x we know $A(x, z') = z$. Thus $z = A(x, z') = A(x, A(x + 1, y))$, which is equal to $A(x + 1, y + 1)$ by definition of A . \square

We now show that the graph $\Gamma \subseteq \omega^3$ of the Ackermann function is primitive recursive. As an auxiliary relation, let $R(u_1, u_2, u_3, k) \subseteq \omega^4$ be the set of all quadruples such that

$$(\beta(u_1, i), \beta(u_2, i), \beta(u_3, i))_{i \leq k}$$

is an Ackermann calculation. R is primitive recursive, since $R(u_1, u_2, u_3, k)$ is equivalent to

$$\forall i \leq k \begin{cases} \beta(u_3, i) = \beta(u_2, i) + 1 & \text{if } \beta(u_1, i) = 0 \\ \exists j \leq k \left(\beta(u_1, j) = \beta(u_1, i \dot{-} i) \wedge \right. & \text{if } \beta(u_1, i) > 0 \wedge \beta(u_2, i) = 0 \\ \quad \left. \beta(u_2, j) = 1 \wedge \beta(u_3, j) = \beta(u_3, i) \right) \\ \exists j, j' \leq k \left(\beta(u_1, j') = \beta(u_1, i) \wedge \right. & \text{if } \beta(u_1, i) > 0 \wedge \beta(u_2, i) > 0 \\ \quad \left. \beta(u_2, j') = \beta(u_2, i) \dot{-} 1 \wedge \right. \\ \quad \left. \beta(u_1, j) = \beta(u_1, i) \dot{-} 1 \wedge \right. \\ \quad \left. \beta(u_2, j) = \beta(u_3, j') \wedge \beta(u_3, j) = \beta(u_3, i) \right) \end{cases}$$

Having confirmed that R is primitive recursive we can now define the graph $\Gamma(x, y, z)$ of the Ackermann function (using the primitive recursive function ρ from 1.3.8(ii)) by

$$\exists u_1, u_2, u_3, i \leq \rho((z + 1)^3) \left(R(u_1, u_2, u_3, (z + 1)^3) \wedge \right. \\ \left. x = \beta(u_1, i) \wedge y = \beta(u_2, i) \wedge z = \beta(u_3, i) \right).$$

Using 1.4.12(i) \Leftrightarrow (iii), this shows that the graph of the Ackermann function is primitive recursive. This finishes the proof of 1.4.8.

1.5. Recursively enumerable sets.

1.5.1. Definition. A subset R of ω^n is called **recursively enumerable** or **computably enumerable** if there is a recursive set $R' \subseteq \omega^n \times \omega$ such that R is the projection of R' onto the first n coordinates. Hence

$$R(a) \iff \exists n \in \omega : R'(a, n).$$

The intuition behind this notion is that a recursively enumerable set is one which can be listed by a machine, i.e. the machine will output every element of R at some point, and it will only output elements of R . However, the machine cannot answer the question on whether a given element is in R : for a given element which is not yet listed at some point, we don't know whether the machine simply has not yet listed that element or whether the machine will never list it (because the element is not in R).

An algorithm that will do the listing can be thought of checking all tuples $(a, n) \in \omega^n \times \omega$ against membership in R' (which is recursive) and then output those a for which the answer to $(a, n) \in R'$ is 'yes'.

On the other hand if we have two machines, one listing the elements of R , the other one listing the elements in the complement of R , then R is recursive:

1.5.2. Negation Theorem

$R \subseteq \omega^n$ is recursive if and only if R and $\neg R$ are recursively enumerable.

Proof. If R is recursive, then R is the projection of $R \times \omega$, which is recursive too. Since complements of recursive sets are again recursive, this shows one implication.

Conversely, suppose R and $\neg R$ are recursively enumerable. Take $R', R'' \subseteq \omega^n \times \omega$ such that R is the projection of R' and $\neg R$ is the projection of R'' onto the first n coordinates. Then R is recursive because the function

$$f(a) = \mu x (R'(a, x) \vee R''(a, x))$$

is recursive and

$$R(a) \iff R'(a, f(a)).$$

□

1.5.3. Proposition. *The following are equivalent for every non-empty subset $R \subseteq \omega^n$.*

- (i) R is recursively enumerable.
- (ii) There are recursive functions $f_1, \dots, f_n : \omega \rightarrow \omega$ such that R is the image of the function

$$(f_1, \dots, f_n) : \omega \rightarrow \omega^n.$$

- (iii) There are some $k \in \mathbb{N}$ and recursive functions $f_1, \dots, f_n : \omega^k \rightarrow \omega$ such that R is the image of the function

$$(f_1, \dots, f_n) : \omega^k \rightarrow \omega^n.$$

Proof. (i) \Rightarrow (ii). Suppose R is equal to the projection of a non-empty recursive set $R' \subseteq \omega^n \times \omega$ to the first n coordinates. Pick $a = (a_1, \dots, a_n) \in R$ and define for $i \in \{1, \dots, n\}$ the function $f_i : \omega \rightarrow \omega$ by

$$f_i(x) = \begin{cases} (x)_i & \text{if } R'((x)_1, \dots, (x)_n, (x)_{n+1}) \\ a_i & \text{otherwise.} \end{cases}$$

The f_i are recursive since R' is recursive and the coordinate functions $(x)_j$ are primitive recursive. R is in the image of $f = (f_1, \dots, f_n)$, because for $(b_1, \dots, b_n, c) \in R'$ there is some $x \in \omega$ with

$$((x)_1, \dots, (x)_n, (x)_{n+1}) = (b_1, \dots, b_n, c)$$

and so $f(x) = (b_1, \dots, b_n)$.

To see that $f(\omega) \subseteq R$, take $x \in \omega$ and assume $f(x) \neq a$. Then for some i we have $f_i(x) \neq a_i$. This is only possible if $R'((x)_1, \dots, (x)_n, (x)_{n+1})$. Consequently

$$f(x) = ((x)_1, \dots, (x)_n) \text{ is in } R.$$

(ii) \Rightarrow (iii) is trivial: Take $k = 1$.

(iii) \Rightarrow (i) Take recursive functions $f_1, \dots, f_n : \omega^k \rightarrow \omega$ such that R is the image of the function $f = (f_1, \dots, f_n) : \omega^k \rightarrow \omega^n$. Let

$$R' = \{(a_1, \dots, a_n, x) \in \omega^n \times \omega \mid \bigwedge_{i=1}^n a_i = f_i((x)_1, \dots, (x)_k)\}.$$

Then R' is recursive and R is the projection of R' onto the first n coordinates. \square

2. FORMAL PROOFS AND THE COMPLETENESS THEOREM

The first two sections contain the revision of predicate logic again, as posted on the website from week 0. Proofs are omitted. A full version of the entire chapter including all proofs may be found at <http://personalpages.manchester.ac.uk/staff/Marcus.Tressl/teaching/Goedel/PredicateLogic.pdf>.

2.1. Languages and formulas.

In this section we shall define what is a first order language, usually denoted by \mathcal{L} . \mathcal{L} will consist of an alphabet and a set of finite sequences (strings) of elements of that alphabet, built according to certain rules; these strings will be called formulas.

The alphabet of a language

2.1.1. Definition. (Alphabet)

The **alphabet** of a language \mathcal{L} consists of the following data:

- (I) A set of **logical symbols**, which are present in every language:
 - \neg ('not'), \rightarrow ('implies'), \forall ('for all')
 - The equality symbol: $=$
 - Brackets: $)$ $($
 - Comma: $,$
 - Symbols to denote **variables**: v_0, v_1, v_2, \dots . Notice that each v_i is considered as a single symbol (and not as a concatenation of two symbols).
- (II)
 - Three mutually disjoint sets \mathcal{R} (called the set of **relation symbols** or **predicate symbols**), \mathcal{F} (called the set of **function symbols**) and \mathcal{C} (called the set of **constant symbols**). Further, none of these sets contains a logical symbol.
 - Maps
 - $\lambda : \mathcal{R} \rightarrow \mathbb{N}$ called the “**arity of relation symbols**”
 - $\mu : \mathcal{F} \rightarrow \mathbb{N}$ called the “**arity of function symbols**”

For $R \in \mathcal{R}$ and $F \in \mathcal{F}$, the numbers $\lambda(R)$ and $\mu(F)$ are called the **arity** of R , F respectively. We say that R , F is **n -ary**, if $\lambda(R) = n$, $\mu(F) = n$, respectively.

Every logical symbol and every element from $\mathcal{R} \cup \mathcal{F} \cup \mathcal{C}$ is called an \mathcal{L} -symbol or simply a **symbol** whenever \mathcal{L} is clear from the context. We shall also use the term **(\mathcal{L} -)letter** instead of (\mathcal{L})-symbol.

We define the **set of variables** as

$$\text{Vbl} := \{v_n \mid n \in \mathbb{N}_0\}.$$

The alphabet of a language \mathcal{L} is called **finite** if \mathcal{R} , \mathcal{F} and \mathcal{C} are finite. Otherwise the alphabet of \mathcal{L} is called **infinite**

The alphabet of a language \mathcal{L} is called **countable** if \mathcal{R} , \mathcal{F} and \mathcal{C} are countable or finite. Otherwise the alphabet of \mathcal{L} is called **uncountable**.

In general, the **cardinality of an alphabet of a language** \mathcal{L} is the cardinality of $\mathcal{R} \cup \mathcal{F} \cup \mathcal{C}$.

Notation. Obviously, the alphabet of a language is uniquely determined by the data in item II of definition 2.1.1. These data are called the **similarity type** of \mathcal{L} . Hence the similarity type of \mathcal{L} is given by

$$(\lambda : \mathcal{R} \longrightarrow \mathbb{N}, \mu : \mathcal{F} \longrightarrow \mathbb{N}, \mathcal{C})$$

Extension of languages. If \mathcal{L} is a language of similarity type $(\lambda : \mathcal{R} \longrightarrow \mathbb{N}, \mu : \mathcal{F} \longrightarrow \mathbb{N}, \mathcal{C})$ and \mathcal{L}' is a language of similarity type $(\lambda' : \mathcal{R}' \longrightarrow \mathbb{N}, \mu' : \mathcal{F}' \longrightarrow \mathbb{N}, \mathcal{C}')$, then we say that \mathcal{L}' **extends** \mathcal{L} and denote this by $\mathcal{L} \subseteq \mathcal{L}'$, if $\mathcal{R} \subseteq \mathcal{R}'$, $\mathcal{F} \subseteq \mathcal{F}'$, $\mathcal{C} \subseteq \mathcal{C}'$ and if the arity functions λ', μ' restrict to the arity functions λ, μ respectively.

2.1.2. *Examples.*

- (i) The empty similarity type. Here $\mathcal{R} = \mathcal{F} = \mathcal{C} = \emptyset$. In the terminology of the propositional logic course, this language corresponds to what is called \mathcal{L}_0 there.
- (ii) The similarity type of a composition (or of an operation): $(\emptyset, \mu : \{\circ\} \longrightarrow \{2\}, \emptyset)$. This means: $\mathcal{R} = \mathcal{C} = \emptyset$ and \mathcal{F} consist of a single element \circ of arity 2: $\mu(\circ) = 2$.
- (iii) The similarity type of groups: $(\emptyset, \mu : \{\circ, {}^{-1}\} \longrightarrow \mathbb{N}, \{e\})$ where $\mu(\circ) = 2$ and $\mu({}^{-1}) = 1$; hence \circ is a binary function symbol (i.e. of arity 2), ${}^{-1}$ is a function symbol of arity 1 and e is a constant symbol.
- (iv) The similarity type of unital rings: $(\emptyset, \mu : \{+, -, \cdot\} \longrightarrow \mathbb{N}, \{0, 1\})$, where $\mu(+)$ and $\mu(\cdot) = 2$ and $\mu(-) = 1$. Hence $-$ is a unary (i.e. 1-ary) and $+, \cdot$ are binary function symbols. 0 and 1 are constants.
- (v) The similarity type of set theory: $(\lambda : \{\in\} \longrightarrow \{2\}, \emptyset, \emptyset)$. Here \in is a binary predicate symbol. Sometimes this similarity type also contains a constant symbol (denoting the empty set).
- (vi) The similarity type of partially ordered sets: $(\lambda : \{\leq\} \longrightarrow \{2\}, \emptyset, \emptyset)$. Here \leq is a binary predicate symbol.
- (vii) The similarity type of ordered groups: $(\lambda : \{\leq\} \longrightarrow \{2\}, \mu : \{\circ, {}^{-1}\} \longrightarrow \mathbb{N}, \{e\})$. Here \leq is a binary relation symbol.

Terms

2.1.3. **Definition.** (\mathcal{L} -term)

Given the similarity type $(\lambda : \mathcal{R} \longrightarrow \mathbb{N}, \mu : \mathcal{F} \longrightarrow \mathbb{N}, \mathcal{C})$ of \mathcal{L} , we define subsets $\text{tm}_k(\mathcal{L})$ of strings (i.e. of finite sequences) of the alphabet of \mathcal{L} by induction on $k \in \mathbb{N}_0$ as follows:

$$\begin{aligned} \text{tm}_0(\mathcal{L}) &= \text{Vbl} \cup \mathcal{C} \text{ and} \\ \text{tm}_{k+1}(\mathcal{L}) &= \text{tm}_k(\mathcal{L}) \cup \{F(t_1, t_2, \dots, t_n) \mid n \in \mathbb{N}, F \in \mathcal{F}, \mu(F) = n, \\ &\quad t_1, \dots, t_n \in \text{tm}_k(\mathcal{L})\}. \end{aligned}$$

The set of \mathcal{L} -**terms** is defined as

$$\text{tm}(\mathcal{L}) := \bigcup_{k \in \mathbb{N}_0} \text{tm}_k(\mathcal{L}).$$

The elements of $\text{tm}(\mathcal{L})$ are called \mathcal{L} -**terms** or simply 'terms' if \mathcal{L} is clear from the context.

The **complexity of an \mathcal{L} -term t** - denoted by $c(t)$ - is the least $k \in \mathbb{N}_0$ such that $t \in \text{tm}_k(\mathcal{L})$. Notice that for $t \in \text{tm}(\mathcal{L})$ and $k \in \mathbb{N}_0$ we have by definition $c(t) \leq k \iff t \in \text{tm}_k(\mathcal{L})$.

2.1.4. Explanation of the definition of a term. Each term is built up inductively, starting from variables and constant symbols, by creating expressions of the form $F(t_1, \dots, t_n)$, where F is an n -ary function symbol and t_1, \dots, t_n are previously constructed terms. Notice that $F(t_1, \dots, t_n)$ is (at the moment) just a list of symbols in our language - F is not a function and one cannot plug anything into it.

We do an example. Suppose \mathcal{L} is the language $\{+, \cdot, 0, 1\}$, where $+$ and \cdot are binary function symbols (hence, they are elements of \mathcal{F} and their arity is $\mu(+) = \mu(\cdot) = 2$), and $0, 1$ are constant symbols (not the numbers $0, 1$). Here are a few terms in that language:

- (a) $0, 1, v_7$. These are three terms of complexity 0.
- (b) $+(0, v_2), \cdot(1, 1), +(v_1, v_3)$. These are three terms of complexity 1. In a human readable form we would write $0 + v_2, 1 \cdot 1, v_1 + v_2$.
- (c) $+(1, \cdot(v_3, v_1)), \cdot(+ (1, v_3), \cdot v_1)$. These are two terms of complexity 2. In a human readable form we might be tempted to write $1 + v_3 \cdot v_1$ and $(1 + v_3) \cdot v_1$; however, there is - a priori - something fishy with the expression $1 + v_3 \cdot v_1$. Can you spot it?
- (d) $\cdot(* (+ (1, 1), v_2), v_2)$ is a term of complexity 3. In human readable form this would be $2v_2^2$.

We can see that in this language, terms (in human readable form) look like polynomials (with coefficients in \mathbb{N}). In fact, if you understand this example well, the general case is in essence not more complicated, just more general.

2.1.5. Theorem. (*Unique readability theorem for terms*) *If t is an \mathcal{L} -term, then either t is a variable or t is a constant symbol or there are uniquely determined $n \in \mathbb{N}$, $F \in \mathcal{F}$ of arity n and $t_1, \dots, t_n \in \text{tm}(\mathcal{L})$ such that*

$$t = F(t_1, t_2, \dots, t_n).$$

The unique readability theorem is important when we make definition or when we prove properties of terms by induction, e.g., see 2.2.2(A).

2.1.6. Corollary. *For all $n \in \mathbb{N}$, all \mathcal{L} -terms t_1, \dots, t_n and each n -ary function symbol F of \mathcal{L} we have*

$$c(F(t_1, \dots, t_n)) = 1 + \max\{c(t_1), \dots, c(t_n)\}.$$

Formulas

2.1.7. Definition. (formulas)

Given a similarity type $(\lambda : \mathcal{R} \longrightarrow \mathbb{N}, \mu : \mathcal{F} \longrightarrow \mathbb{N}, \mathcal{C})$ of a language \mathcal{L} , an **atomic \mathcal{L} -formula** is a string of the alphabet of \mathcal{L} of the form

$$t_1 \doteq t_2,$$

where t_1, t_2 are \mathcal{L} -terms or

$$R(t_1, \dots, t_n),$$

where R is a relation symbol of arity $n \in \mathbb{N}$ and t_1, \dots, t_n are \mathcal{L} -terms. The set of atomic \mathcal{L} -formulas is denoted by $\text{at-Fml}(\mathcal{L})$.

We define

$$\begin{aligned} \text{Fml}_0(\mathcal{L}) &= \text{at-Fml}(\mathcal{L}) \text{ and inductively for each } k \in \mathbb{N}_0 : \\ \text{Fml}_{k+1}(\mathcal{L}) &= \text{Fml}_k(\mathcal{L}) \cup \{(\neg\varphi), (\varphi \rightarrow \psi), (\forall x\varphi) \mid \varphi, \psi \in \text{Fml}_k(\mathcal{L}), x \in \text{Vbl}\}. \end{aligned}$$

The set of \mathcal{L} -**formulas** is defined as

$$\text{Fml}(\mathcal{L}) := \bigcup_{k \in \mathbb{N}_0} \text{Fml}_k(\mathcal{L}).$$

If the letter \forall does not occur in the \mathcal{L} -formula φ , then φ is called **quantifier free**.

Warning. Not every formula that has (obvious) meaning in mathematics is a formula in our sense. This is in particular important after we have proved significant theorems involving formulas. Here is an example:

$$\forall n \in \mathbb{N} \exists r, q \in \mathbb{N}_0 \ n = q \cdot m + r \wedge r < m.$$

There is no language (according to our definition) such that the above is a formula in that language.

Notice that the quantifier introduced in the definition of $\text{Fml}_{k+1}(\mathcal{L})$ (cf. 2.1.7) is always applied in a nonrestricted way, e.g.

$$\forall n \exists r, q \ n \doteq q \cdot m + r \wedge r < m$$

will be a formula in the language of rings; we introduce the appropriate abbreviations (concerning the symbols \exists and \wedge) shortly.

The **language** or **signature** \mathcal{L} is the triple consisting of the alphabet of \mathcal{L} , the set of \mathcal{L} -terms and the set of \mathcal{L} -formulas. Obviously, $\text{tm}(\mathcal{L})$ and $\text{Fml}(\mathcal{L})$ are uniquely determined by the similarity type of \mathcal{L} and we shall simply communicate languages by their similarity type.

Hence the expression 'let $\mathcal{L} = (\lambda : \mathcal{R} \rightarrow \mathbb{N}, \mu : \mathcal{F} \rightarrow \mathbb{N}, \mathcal{C})$ be a language' stands for 'let \mathcal{L} be the language with similarity type $(\lambda : \mathcal{R} \rightarrow \mathbb{N}, \mu : \mathcal{F} \rightarrow \mathbb{N}, \mathcal{C})$ '.

We say that a language is **finite**, **infinite**, **countable** or **uncountable** if the **alphabet** of that language has this property. In general, the cardinality of a language \mathcal{L} , denoted by $\text{card}(\mathcal{L})$, is the cardinality of the **alphabet** of that language.

2.1.8. Lemma. *The cardinality of $\text{Fml}(\mathcal{L})$ is the maximum of \aleph_0 and the cardinality of \mathcal{L} . If \mathcal{L} is countable, then the sets $\text{tm}(\mathcal{L})$ and $\text{Fml}(\mathcal{L})$ are countable and infinite.*

As for terms we have a unique readability theorem:

2.1.9. Theorem. *(Unique readability theorem for formulas)*

Let $\mathcal{L} = (\lambda : \mathcal{R} \rightarrow \mathbb{N}, \mu : \mathcal{F} \rightarrow \mathbb{N}, \mathcal{C})$ be a language and let φ be an \mathcal{L} -formula. Then exactly one of the following holds true:

- (i) φ is atomic and there are uniquely determined $t_1, t_2 \in \text{tm}(\mathcal{L})$ such that φ is $t_1 \doteq t_2$, or
- (ii) φ is atomic and there are a unique $n \in \mathbb{N}$, a unique $R \in \mathcal{R}$ and uniquely determined \mathcal{L} -terms t_1, \dots, t_n such that φ is $R(t_1, \dots, t_n)$, or
- (iii) φ is equal to a string of the form $(\neg\psi)$ for a uniquely determined $\psi \in \text{Fml}(\mathcal{L})$, or

- (iv) φ is equal to a string of the form $(\varphi_1 \rightarrow \varphi_2)$ for uniquely determined $\varphi_1, \varphi_2 \in \text{Fml}(\mathcal{L})$, or
 (v) φ is equal to a string of the form $(\forall x\psi)$ for uniquely determined $\psi \in \text{Fml}(\mathcal{L})$ and $x \in \text{Vbl}$.

Domestication of the notation

- We will omit brackets if this does not lead to ambiguity.
- We use the following abbreviation for \mathcal{L} -formulas φ, ψ : $\varphi \vee \psi := (\neg\varphi) \rightarrow \psi$, $\varphi \wedge \psi := \neg(\varphi \rightarrow (\neg\psi))$, $\varphi \leftrightarrow \psi := (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$ and $\exists x\varphi := \neg\forall x(\neg\varphi)$ where x is a variable.
- We write

$\forall x_1, \dots, x_n \varphi$ instead of $\forall x_1 \dots \forall x_n \varphi$ and $\exists x_1, \dots, x_n \varphi$ instead of $\exists x_1 \dots \exists x_n \varphi$

where each x_i is a variable.

The strings $\forall x$ and $\exists x$ are called **quantifiers**. A string of quantifiers is a string of the form $Q_1 x_1 \dots Q_n x_n$, where each Q_i is either \forall or \exists and each x_i is a variable.

- We write

$\bigwedge_{i=1}^n \varphi_i$ instead of $\overbrace{(\dots (\varphi_1 \wedge \varphi_2) \wedge \varphi_3) \dots \wedge \varphi_n}$ and

$\bigvee_{i=1}^n \varphi_i$ instead of $\overbrace{(\dots (\varphi_1 \vee \varphi_2) \vee \varphi_3) \dots \vee \varphi_n}$.

- We write $t_1 \neq t_2$ instead of $(\neg t_1 \doteq t_2)$.
- If R is a binary relation, we write $t_1 R t_2$ instead of $R(t_1, t_2)$.

Complexity and subformulas

The unique readability theorems 2.1.5 and 2.1.9 allow us to define new objects from formulas, and to prove statements about formulas. This will be done via induction on the construction depth (or the 'complexity') of terms and formulas:

2.1.10. Definition. The **complexity of an \mathcal{L} -formula φ** - denoted by $c(\varphi)$ - is the least $k \in \mathbb{N}_0$ such that $\varphi \in \text{Fml}_k(\mathcal{L})$.

Notice that this is not in conflict with the definition of the complexity of \mathcal{L} -terms (cf. 2.1.3), since the set of \mathcal{L} -terms is disjoint from the set of \mathcal{L} -formulas. Notice also that for any given terms t_1, t_2, \dots, t_n and each n -ary relation symbol R of \mathcal{L} , $c(R(t_1, \dots, t_n)) = 0$. Similarly $c(t_1 \doteq t_2) = 0$.

By definition, for every \mathcal{L} -formula φ and each $k \in \mathbb{N}_0$ we have

$$c(\varphi) \leq k \iff \varphi \in \text{Fml}_k(\mathcal{L}).$$

2.1.11. Lemma. For all \mathcal{L} -formulas φ, ψ we have

$$c(\neg\varphi) = 1 + c(\varphi), \quad c(\varphi \rightarrow \psi) = 1 + \max\{c(\varphi), c(\psi)\} \quad \text{and} \quad c(\forall x\varphi) = 1 + c(\varphi).$$

2.1.12. Definition. (subformula)

We define a binary relation between \mathcal{L} -formulas φ and ψ

- called "**... is a subformula of ...**" -

inductively, w.r.t. the complexity of ψ :

- (i) If $c(\psi) = 0$ (equivalently: ψ is atomic), then $\varphi = \psi$.
- (ii) If $c(\psi) = k + 1$, then
 - (a) If $\psi = (\forall x\vartheta)$ or $\psi = (\neg\vartheta)$, then φ is a subformula of ϑ or $\varphi = \psi$
 - (b) If $\psi = (\psi_1 \rightarrow \psi_2)$, then φ is a subformula of ψ_1 or φ is a subformula of ψ_2 or $\varphi = \psi$.

Notice that this definition is correct by 2.1.11.

Of course, every subformula of φ occurs in φ at some position. Also, note that by a straightforward induction on the complexity, we see that the subformula relation is transitive and a formula φ is a subformula of ψ is an only if φ occurs as a substring of ψ .

Free and bound occurrences of variables

Let $\mathcal{L} = (\lambda : \mathcal{R} \rightarrow \mathbb{N}, \mu : \mathcal{F} \rightarrow \mathbb{N}, \mathcal{C})$ be a language.

2.1.13. **Definition.** (scope of a quantifier)

The **scope** of a quantifier $\forall x$ in an \mathcal{L} -formula φ is the set of all positions of letters in φ , which are captured in a subformula of the form $(\forall x\psi)$ of φ .

More formally: the scope of $\forall x$ in φ is the set of all $k \in \mathbb{N}$ such that there is a subformula of the form $(\forall x\psi)$ of φ , of length $l \in \mathbb{N}$ which occurs at a position p in φ with $p \leq k < p + l$.

2.1.14. *Example.* For example, look at the formula of the language of ordered groups (cf. 2.1.2):

$$\varphi = (\forall v_2((\forall v_1 \circ (e, v_1) \doteq v_5) \rightarrow \neg(v_1 \leq e))).$$

Here the scope of the quantifier $\forall v_1$ in φ :

$$\varphi = (\forall v_2(\overbrace{(\forall v_1 \circ (e, v_1) \doteq v_5)}^{\text{scope of } \forall v_1} \rightarrow \neg(v_1 \leq e))).$$

2.1.15. **Definition.** (free and bound occurrence of variables)

Let φ be an \mathcal{L} -formula and let x be a variable.

- (i) If x occurs in φ at position $k \in \mathbb{N}$ and if k is not in the scope of the quantifier $\forall x$ in φ , then we say x occurs **free** in φ at position k .
- (ii) If x occurs in φ at position $k \in \mathbb{N}$ and if k is in the scope of the quantifier $\forall x$ in φ , then we say x occurs **bound** in φ at position k .
- (iii) x is a **free variable** of φ if there is some $k \in \mathbb{N}$ such that x occurs free in φ at position k .

The set of free variables of φ is denoted by $\text{Fr}(\varphi)$. If $\text{Fr}(\varphi) = \emptyset$, then φ is called an (\mathcal{L} -)**sentence** and the set of all \mathcal{L} -sentences is denoted by $\text{Sen}(\mathcal{L})$.

It is convenient to extend the notation to terms:

- (iv) If t is an \mathcal{L} -term, then we define $\text{Fr}(t)$ to be the set of all variables occurring in t and we will also say that x is free in t instead of $x \in \text{Fr}(t)$. Notice that there are no variables which are possibly bound in t . If $\text{Fr}(t) = \emptyset$, then t is called a **closed term** or a **constant term**.

Different occurrences of a given variable in a formula may be free or bound, depending on where they are. In example 2.1.14 above, v_1 occurs bound at two positions in φ and free at one position.

$$\varphi = (\forall v_2((\forall \overbrace{v_1}^{\text{bound occurrence}} \circ (e, \overbrace{v_1}^{\text{bound occurrence}}) \doteq v_5) \rightarrow \neg(\overbrace{v_1}^{\text{free occurrence}} \leq e))).$$

We have $\text{Fr}(\varphi) = \{v_1, v_5\}$.

2.1.16. Lemma. *Let φ, ψ be \mathcal{L} -formulas.*

- (i) *If φ is quantifier free then $\text{Fr}(\varphi)$ is the set of variables occurring in φ .*
- (ii) $\text{Fr}(\neg\varphi) = \text{Fr}(\varphi)$
- (iii) $\text{Fr}((\varphi \rightarrow \psi)) = \text{Fr}(\varphi) \cup \text{Fr}(\psi)$.
- (iv) $\text{Fr}(\forall x\varphi) = \text{Fr}(\varphi) \setminus \{x\}$ for all $x \in \text{Vbl}$.

2.1.17. Notation.

- The expressions ' $t(x_1, \dots, x_n) \in \text{tm}(\mathcal{L})$ ' or 'let $t(x_1, \dots, x_n)$ be an \mathcal{L} -term' are shorthand for
 “ $t \in \text{tm}(\mathcal{L}), x_1, \dots, x_n \in \text{Vbl}$ with $x_i \neq x_j$ ($i \neq j$) and $\text{Fr}(t) \subseteq \{x_1, \dots, x_n\}$ ”.
- The expressions ' $\varphi(x_1, \dots, x_n) \in \text{Fml}(\mathcal{L})$ ' or 'let $\varphi(x_1, \dots, x_n)$ be an \mathcal{L} -formula' are shorthand for
 “ $\varphi \in \text{Fml}(\mathcal{L}), x_1, \dots, x_n \in \text{Vbl}$ with $x_i \neq x_j$ ($i \neq j$) and $\text{Fr}(\varphi) \subseteq \{x_1, \dots, x_n\}$ ”.

2.1.18. Definition. Let φ be an \mathcal{L} -formula.

- (i) Let x, y be variables. We define

x is free in φ for y or y is substitutable for x in φ

if no position of φ at which x occurs free in φ , is in the scope of the quantifier $\forall y$ in φ .

- (ii) Let t be an \mathcal{L} -term. We define

x is free in φ for t or t is substitutable for x in φ

if x is free in φ for every variable which occurs in t .

So by definition, each variable x is free for x in φ and each variable which does not occur in φ is free in φ for every term.

In example 2.1.14, i.e.

$$\varphi = (\forall v_2((\forall v_1 \circ (e, v_1) \doteq v_5) \rightarrow \neg^1(v_1) \leq e)),$$

v_1 is free for v_5 but not free for v_2 in φ ; v_5 is not free for the term $\circ(v_2, v_5)$.

2.1.19. Definition. Let $\varphi \in \text{Fml}(\mathcal{L})$, $t_1, \dots, t_n, t \in \text{tm}(\mathcal{L})$ and let x_1, \dots, x_n be n distinct variables.

- (i) The expression $t(x_1/t_1, \dots, x_n/t_n)$ denotes the string obtained from t by replacing every occurrence of x_i in t with the string t_i ($1 \leq i \leq n$).
- (ii) If for each $i \in \{1, \dots, n\}$ the variable x_i is free in φ for t_i then the expression $\varphi(x_1/t_1, \dots, x_n/t_n)$ denotes the string obtained from φ by simultaneously replacing every free occurrence of x_i in φ with the string t_i ($1 \leq i \leq n$). We call $\varphi(x_1/t_1, \dots, x_n/t_n)$ the **substitution** of x_1, \dots, x_n by t_1, \dots, t_n in φ .

Warning. Notice that we replace the variables x_i by the terms t_i simultaneously and not consecutively: For example if φ is $(\forall x_2 x_1 \doteq x_2) \rightarrow x_2 \doteq x_3$, then $\varphi(x_1/t_1, x_2/t_2)$ is $(\forall x_2 t_1 \doteq x_2) \rightarrow t_2 \doteq x_3$.

However, in general $\varphi(x_1/t_1, x_2/t_2)$ is NOT the same as $\varphi(x_1/t_1)(x_2/t_2)$. Why?

2.1.20. Lemma. *Let $\varphi \in \text{Fml}(\mathcal{L})$, $t_1, \dots, t_n, t \in \text{tm}(\mathcal{L})$ and let x_1, \dots, x_n be n distinct variables.*

(i) $t(x_1/t_1, \dots, x_n/t_n)$ is an \mathcal{L} -Term and if $\text{Fr}(t) \subseteq \{x_1, \dots, x_n\}$, then

$$\text{Fr}(t(x_1/t_1, \dots, x_n/t_n)) \subseteq \text{Fr}(t_1) \cup \dots \cup \text{Fr}(t_n).$$

(ii) If for each $i \in \{1, \dots, n\}$ the variable x_i is free in φ for t_i then the string $\varphi(x_1/t_1, \dots, x_n/t_n)$ is an \mathcal{L} -formula and in the case $\text{Fr}(\varphi) \subseteq \{x_1, \dots, x_n\}$ we have

$$\text{Fr}(\varphi) \subseteq \text{Fr}(t_1) \cup \dots \cup \text{Fr}(t_n).$$

2.2. Structures and Tarski's definition of truth.

Throughout, $\mathcal{L} = (\lambda : \mathcal{R} \rightarrow \mathbb{N}, \mu : \mathcal{F} \rightarrow \mathbb{N}, \mathcal{C})$ denotes a formal language.

2.2.1. **Definition.** An \mathcal{L} -**structure** is a tuple

$$\mathcal{M} = \left(M, (R^{\mathcal{M}} \mid R \in \mathcal{R}), (F^{\mathcal{M}} \mid F \in \mathcal{F}), (c^{\mathcal{M}}, c \in \mathcal{C}) \right)$$

consisting of

(S1) A nonempty set M , called the **universe** or the **domain** or the **carrier** of \mathcal{M} . We shall also write $|\mathcal{M}|$ instead of M .

(S2) A family $(R^{\mathcal{M}} \mid R \in \mathcal{R})$ of relations of M such that for $R \in \mathcal{R}$, $R^{\mathcal{M}} \subseteq M^{\lambda(R)}$. Hence $R^{\mathcal{M}}$ is a $\lambda(R)$ -ary relation of M , called the **interpretation of R in \mathcal{M}** . Observe that for different $R_1, R_2 \in \mathcal{R}$ we may have $R_1^{\mathcal{M}} = R_2^{\mathcal{M}}$. Formally, $(R^{\mathcal{M}} \mid R \in \mathcal{R})$ is a map $\mathcal{R} \rightarrow \bigcup_{n \in \mathbb{N}} \mathcal{P}(M^n)$ such that the image $R^{\mathcal{M}}$ of $R \in \mathcal{R}$ under this map is a subset of $M^{\lambda(R)}$.

(S3) A family $(F^{\mathcal{M}} \mid F \in \mathcal{F})$ of functions, where for $F \in \mathcal{F}$, $F^{\mathcal{M}} : M^{\mu(F)} \rightarrow M$. Hence $F^{\mathcal{M}}$ is a $\mu(F)$ -ary function of M , called the **interpretation of F in \mathcal{M}** . Observe that for different $F_1, F_2 \in \mathcal{F}$ we may have $F_1^{\mathcal{M}} = F_2^{\mathcal{M}}$. Formally, $(F^{\mathcal{M}} \mid F \in \mathcal{F})$ is a map $\mathcal{F} \rightarrow \bigcup_{n \in \mathbb{N}} \text{Maps}(M^n, M)$ such that the image $F^{\mathcal{M}}$ of $F \in \mathcal{F}$ under this map is a function $M^{\mu(F)} \rightarrow M$.

(S4) A family $(c^{\mathcal{M}} \mid c \in \mathcal{C})$ of elements of M . Hence $c^{\mathcal{M}}$ is an element of M , called the **interpretation of c in \mathcal{M}** . Observe that for different $c_1, c_2 \in \mathcal{C}$ we may have $c_1^{\mathcal{M}} = c_2^{\mathcal{M}}$.

Formally, $(c^{\mathcal{M}} \mid c \in \mathcal{C})$ is simply a map $\mathcal{C} \rightarrow M$.

\mathcal{M} is called **finite/countable/uncountable/infinite** if its universe $|\mathcal{M}|$ is finite/countable/uncountable/infinite. More generally, the **size of a structure \mathcal{M}** is the cardinality $\text{card}(|\mathcal{M}|)$ of its universe.

2.2.2. **Definition.** Let \mathcal{M} be an \mathcal{L} -structure with domain $M = |\mathcal{M}|$.

(A) We define by induction on the complexity of an \mathcal{L} -term $t(x_1, \dots, x_n)$ and elements $a_1, \dots, a_n \in M$, an element $t^{\mathcal{M}}(a_1, \dots, a_n) \in M$ as follows:

(i) If $c(t) = 0$, then

$$t^{\mathcal{M}}(a_1, \dots, a_n) = \begin{cases} c^{\mathcal{M}} & \text{if } t \text{ is } c \in \mathcal{C} \\ a_i & \text{if } t \text{ is } v_i \in \text{Vbl}. \end{cases}$$

(ii) If t_1, \dots, t_n are \mathcal{L} -terms and $F \in \mathcal{F}$ with $\mu(F) = n$, then we define $F(t_1, \dots, t_n)^{\mathcal{M}}(a_1, \dots, a_n) := F^{\mathcal{M}}(t_1^{\mathcal{M}}(a_1, \dots, a_n), \dots, t_n^{\mathcal{M}}(a_1, \dots, a_n))$.

This is a correct definition by 2.1.5.

(B) We define by induction on the complexity of an \mathcal{L} -formula $\varphi(x_1, \dots, x_n)$ and all $a_1, \dots, a_n \in M$ the expression $\varphi(a_1, \dots, a_n)$ **holds in** \mathcal{M} , or \mathcal{M} **satisfies** $\varphi(a_1, \dots, a_n)$, denoted by

$$\mathcal{M} \models \varphi(a_1, \dots, a_n),$$

as follows:

(i) If φ is of the form $t_1 \doteq t_2$ with \mathcal{L} -terms t_1, t_2 then

$$\mathcal{M} \models \varphi(a_1, \dots, a_n) \iff t_1^{\mathcal{M}}(a_1, \dots, a_n) = t_2^{\mathcal{M}}(a_1, \dots, a_n).$$

If φ is of the form $R(t_1, \dots, t_k)$ with $R \in \mathcal{R}$ of arity k and $t_1, \dots, t_k \in \text{tm}(\mathcal{L})$ then

$$\mathcal{M} \models \varphi(a_1, \dots, a_n) \iff (t_1^{\mathcal{M}}(a_1, \dots, a_n), \dots, t_k^{\mathcal{M}}(a_1, \dots, a_n)) \in R^{\mathcal{M}}.$$

(ii) For the induction step we take $\varphi, \psi \in \text{Fml}(\mathcal{L})$, $x \in \text{Vbl}$ and define

$$\begin{aligned} \bullet \mathcal{M} \models (\varphi \rightarrow \psi)(a_1, \dots, a_n) &\iff \\ &\text{if } \mathcal{M} \models \varphi(a_1, \dots, a_n) \text{ then } \mathcal{M} \models \psi(a_1, \dots, a_n), \end{aligned}$$

$$\begin{aligned} \bullet \mathcal{M} \models (\neg\varphi)(a_1, \dots, a_n) &\iff \\ \mathcal{M} \not\models \varphi(a_1, \dots, a_n) &\text{ i.e. } \mathcal{M} \models \varphi(a_1, \dots, a_n) \text{ does not hold} \end{aligned}$$

and

$$\begin{aligned} \bullet \mathcal{M} \models (\forall y\varphi)(a_1, \dots, a_n) &\iff \\ \begin{cases} \mathcal{M} \models (\forall y\varphi)(a_1, \dots, a_n), & \text{if } y \text{ is not among the } x_i \\ \mathcal{M} \models \varphi(a_1, \dots, \underbrace{b}_{i^{\text{th}} \text{ position}}, \dots, a_n) & \text{for all } b \in |\mathcal{M}|, \text{ if } y = x_i. \end{cases} \end{aligned}$$

This is a correct definition by 2.1.9.

(C) Let Σ and Ψ be sets of \mathcal{L} -formulas in at most n free variables x_1, \dots, x_n . Let $\bar{a} = (a_1, \dots, a_n) \in M$. \mathcal{M} is called a **model of** Σ **at** \bar{a} if

$$\mathcal{M} \models \sigma(\bar{a}) \text{ for all } \sigma \in \Sigma.$$

We denote this by

$$\mathcal{M} \models \Sigma(\bar{a}).$$

In particular, if Σ is a set of \mathcal{L} -sentences, so here $n = 0$, it makes sense to write

$$\mathcal{M} \models \Sigma$$

and to say that \mathcal{M} is a model of Σ .

We say that Σ has a model if it has a model \mathcal{M} at some n -tuple with entries in $|\mathcal{M}|$. In this case, Σ is called **satisfiable**.

We say that Σ **logically implies** Φ and write $\Sigma \models \Phi$ if for every \mathcal{L} structure \mathcal{M} and all $a_1, \dots, a_n \in |\mathcal{M}|$ we have

$$\mathcal{M} \models \Sigma(\bar{a}) \implies \mathcal{M} \models \Phi(\bar{a}).$$

2.3. Logical axioms and the definition of a formal proof. Throughout, $\mathcal{L} = (\lambda : \mathcal{R} \rightarrow \mathbb{N}, \mu : \mathcal{F} \rightarrow \mathbb{N}, \mathcal{C})$ denotes a formal language.

2.3.1. Definition. Each of the following \mathcal{L} -formulas are called **logical Axioms** (of \mathcal{L}), where φ, ψ and γ are \mathcal{L} -formulas:

(AxProp):

- (a) $\varphi \rightarrow (\psi \rightarrow \varphi)$
- (b) $(\varphi \rightarrow (\psi \rightarrow \gamma)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \gamma))$
- (c) $((\neg\psi) \rightarrow (\neg\varphi)) \rightarrow (((\neg\psi) \rightarrow \varphi) \rightarrow \psi)$

(Ax $\forall \rightarrow$): $\forall x(\varphi \rightarrow \psi) \rightarrow (\forall x\varphi \rightarrow \forall x\psi)$

(AxSubst): $(\forall x\varphi) \rightarrow \varphi(x/t)$, where x is free in φ for $t \in \text{tm}(\mathcal{L})$.

(AxGen): $\varphi \rightarrow \forall x\varphi$, where x is not free in φ .

(AxEq): For every \mathcal{L} -term t , every n -ary relation symbol R and all variables x_1, \dots, x_n, x, y, z the axioms

- (a) $x \doteq x$
- (b) $x \doteq y \wedge y \doteq z \rightarrow z \doteq x$
- (c) $x \doteq y \rightarrow t(z/y) \doteq t(z/x)$
- (d) $x \doteq y \rightarrow \left(R(x_1, \dots, x_n)(z/x) \leftrightarrow (R(x_1, \dots, x_n)(z/y)) \right)$

(Ax \forall): Any formula of the form

$$\forall x_1 \dots x_n \varphi,$$

where φ is one of the formulas introduced by the other logical axiom schemes above and $x_1, \dots, x_n \in \text{Vbl}$.

Notice: The axioms above are not examinable. However the next definition is examinable.

2.3.2. Definition. Let $\Sigma \subseteq \text{Fml}(\mathcal{L})$. A **formal proof** or a **deduction** from Σ (in \mathcal{L}) is a finite sequence $(\varphi_1, \dots, \varphi_n)$ of \mathcal{L} -formulas such that for each $k \in \{1, \dots, n\}$ one of the following conditions hold:

(PR1): φ_k is a logical axiom (of \mathcal{L}) or

(PR2): $\varphi_k \in \Sigma$ or

(PR3): (Modus Ponens) There are $i, j < k$ such that

$$\varphi_j \text{ is the formula } \varphi_i \rightarrow \varphi_k.$$

If $\Phi \subseteq \text{Fml}(\mathcal{L})$, then we say Σ **proves** Φ and write

$$\Sigma \vdash_{\mathcal{L}} \Phi \text{ (or simply } \Sigma \vdash \Phi \text{ when } \mathcal{L} \text{ is clear from the context),}$$

if every $\varphi \in \Phi$ is an entry of a proof from Σ . If $\Phi = \{\varphi\}$ we just write $\Sigma \vdash \varphi$. If $\Sigma = \emptyset$ we just write $\vdash \Phi$.

We say that a set of \mathcal{L} -formulas Σ is **consistent** if there is a formula that is **not** provable from Σ . By elementary propositional logic this is equivalent to saying that $\Sigma \not\vdash \varphi \wedge \neg\varphi$ for every \mathcal{L} -formula φ . Equivalently: Σ is consistent if and only if $\Sigma \not\vdash \varphi \wedge \neg\varphi$ for some \mathcal{L} -formula φ .

A set Σ of \mathcal{L} -formulas that is not consistent is called **inconsistent**.

2.3.3. *Remark.* The following are immediate consequences of definition 2.3.2:

- (i) If $(\varphi_1, \dots, \varphi_n)$ is a proof from Σ , then also $(\varphi_1, \dots, \varphi_m)$ is a proof from Σ for every $m \leq n$. Moreover $\varphi_1 \in \Sigma$ or φ_1 is a logical axiom.
- (ii) If $(\varphi_1, \dots, \varphi_n)$ and (ψ_1, \dots, ψ_m) are proofs from Σ , then also $(\varphi_1, \dots, \varphi_n, \psi_1, \dots, \psi_m)$ is a proof from Σ .
- (iii) $\Sigma \vdash \Phi \iff \Sigma \vdash \varphi$ for all $\varphi \in \Phi$.
- (iv) $\Sigma \vdash \Phi$ for all $\Phi \subseteq \Sigma$.
- (v) If $\Sigma \vdash \varphi \rightarrow \psi$ and $\Sigma \vdash \varphi$ then $\Sigma \vdash \psi$.

Proof. The proof here is the same as the corresponding argument in propositional logic and is direct from definition 2.3.2. To get up to speed with these type of arguments, please carry this out. \square

We say $(\varphi_1, \dots, \varphi_n)$ is a **proof of φ from Σ** if $(\varphi_1, \dots, \varphi_n)$ is a proof from Σ and $\varphi = \varphi_n$.

The following theorem follows again immediately from definition 2.3.2 of a formal proof. It is of central importance to Predicate Logic and used in many places later on.

2.3.4. **Theorem.** “proofs are finite”

For every $\varphi \in \text{Fml}(\mathcal{L})$ and all subsets $\Sigma \subseteq \text{Fml}(\mathcal{L})$ the following are equivalent:

- (i) $\Sigma \vdash \varphi$
- (ii) there is a finite subset $\Sigma_0 \subseteq \Sigma$ with $\Sigma_0 \vdash \varphi$
- (iii) there is a proof $(\varphi_1, \dots, \varphi_n)$ of φ from Σ .

Proof. Please carry this out and use it to for practicing definition 2.3.2. \square

2.4. Soundness and the Completeness Theorem.

2.4.1. Soundness Theorem

For any language \mathcal{L} and all $\Sigma, \Phi \subseteq \text{Fml}(\mathcal{L})$, if $\Sigma \vdash \Phi$, then $\Sigma \models \Phi$.

Proof. It is enough to show by induction on n the following: If $(\varphi_1, \dots, \varphi_n)$ is a proof from Σ then $\mathcal{M} \models \varphi_n[h]$ for every assignment h of every \mathcal{L} -structure \mathcal{M} satisfying $\mathcal{M} \models \Sigma[h]$.

If $\varphi_n \in \Sigma$ or φ_n is a logical axiom, then this is checked by inspection. You are invited to carry this out. In particular the claim holds for $n = 1$. Moreover for the induction step “ $n-1 \Rightarrow n$ ” it is clear that we only need to show the claim in the case where φ_n is the result of applying Modus Ponens to two entries of $(\varphi_1, \dots, \varphi_{n-1})$. Hence there are $k, j < n$ such that $\varphi_k = \varphi_j \rightarrow \varphi_n$.

If h is an assignment of \mathcal{M} with $\mathcal{M} \models \Sigma[h]$, then by induction we know $\mathcal{M} \models \varphi_j[h]$ and $\mathcal{M} \models \varphi_j \rightarrow \varphi_n [h]$. By 2.2.2(B)(ii) we get $\mathcal{M} \models \varphi_n[h]$ as desired. \square

2.4.2. Completeness Theorem (Gödel 1929)

For any language \mathcal{L} and all $\Sigma, \Phi \subseteq \text{Fml}(\mathcal{L})$, if $\Sigma \models \Phi$, then $\Sigma \vdash \Phi$.

Proof. [not examinable] This is done in every book on Mathematical Logic and is much harder than the implication in 2.4.1. A full proof that fits precisely to our set up may be found in <http://personalpages.manchester.ac.uk/staff/Marcus.Tressl/teaching/Goedel/PredicateLogic.pdf>. A good explanation (with different notations and an alternative proof system) may be found at <https://plato.stanford.edu/entries/logic-classical/>. \square

2.4.3. The terminology “Completeness Theorem” here has to be understood as follows: By the soundness theorem, our proof system from section 1 is sound. In particular, all \mathcal{L} -axioms and all formulas obtained by writing down proofs (from \emptyset) according to the rules in 2.3.2 are true in all \mathcal{L} -structures (which means they are true at all assignments of all \mathcal{L} -structures). However, why are we not writing down more axioms and proof rules, in order to strengthen what can be formally proved? As long as we can show that these new axioms and rules produce consequences that are true in all \mathcal{L} -structures, we might enhance our proof system. However, the problem does not go away if we do that: We can then just ask the same question for the new system.

Here is where the Completeness Theorem enters the scene. A particular instance of 2.4.2, namely the case $\Sigma = \emptyset$, says that every \mathcal{L} -formula that is true in all \mathcal{L} -structures can indeed be proved with our proof system. Therefore, we do not need to add any axioms or rules. In this sense our proof system is complete, explaining the terminology “Completeness Theorem”.

A particularly important instance of the Completeness Theorem is

2.4.4. Corollary. *Every consistent set Σ of \mathcal{L} -formulas is satisfiable (i.e. has a model at some assignment).*

Proof. Apply 2.4.2, using any inconsistent set Φ . \square

Also observe that every satisfiable set of \mathcal{L} -formulas is consistent, which simply says that our proof system is sound. The strength of the completeness theorem lies in the claim that a model of $\Sigma \subseteq \text{Fml}(\mathcal{L})$ exists as soon as we know that the proof system cannot derive a contradiction from Σ . This in fact is the main task in the

proof of the Completeness Theorem: The Corollary easily implies 2.4.2 - this is a routine reasoning just using arguments from propositional logic.

To appreciate the power of the completeness theorem, look at the following proof of the

2.4.5. Compactness Theorem (of first order predicate logic)

If $\Sigma \subseteq \text{Fml}(\mathcal{L})$ and every finite subset of Σ has a model at some assignment, then also Σ has a model at some assignment.

Proof. By 2.4.4 it is enough to show that Σ is consistent. Since proofs are finite, this can be checked by looking at finite subsets of Σ . Now, our assumption and the Soundness Theorem imply that every finite subset of Σ is consistent. \square

2.5. Propositional Tautologies and the Prenex Normal Form.

The material here is not explicitly examinable, but the statements are needed to understand arguments later on. For example it is sometimes very useful to know that we may assume that all formulas are in prenex normal form (up to logical equivalence), see 2.5.6. So if one does not know that theorem it may be difficult to answer some questions on the example sheets.

2.5.1. Proposition. (Propositional Tautologies)

Let α be a formula of propositional logic and let $n \in \mathbb{N}$ be such that every atomic formula of the propositional calculus that occurs in α is among A_0, \dots, A_n .

Let $\varphi_0, \dots, \varphi_n$ be \mathcal{L} -formulas and let ψ be the \mathcal{L} -string obtained from α by replacing for each $i \in \{0, \dots, n\}$ the letter A_i in α with the string φ_i .

Then ψ is again an \mathcal{L} -formula and if α is a tautology of propositional logic, then

$$\vdash_{\mathcal{L}} \psi.$$

Sketch of the proof. That $\psi \in \text{Fml}(\mathcal{L})$ follows by induction on the construction of ψ using the definition of formulas in propositional logic and in predicate logic.

If α is a tautology, then by the completeness theorem for propositional logic we know that there is a formal proof of α in propositional logic from \emptyset . Now by inspection we see that each formal proof from \emptyset in propositional logic translates into a formal proof from \emptyset from predicate logic when we replace propositional variables by \mathcal{L} -formulas. \square

2.5.2. Example. If $\Sigma \subseteq \text{Fml}(\mathcal{L})$ and $\varphi_1, \dots, \varphi_n \in \text{Fml}(\mathcal{L})$, then

$$\Sigma \vdash \varphi_1 \wedge \dots \wedge \varphi_n \iff \Sigma \vdash \varphi_1 \text{ and } \dots \text{ and } \Sigma \vdash \varphi_n.$$

The implication “ \Rightarrow ” is obtained from 2.5.1 and the fact that $A_1 \wedge \dots \wedge A_n \rightarrow A_i$ is a propositional tautology. The implication “ \Leftarrow ” is an application of 2.5.1 using the corresponding statement in propositional logic.

2.5.3. Definition. An \mathcal{L} -formula φ is said to be in **prenex normal form** if there are $n \in \mathbb{N}_0$, $x_1, \dots, x_n \in \text{Vbl}$, $Q_1, \dots, Q_n \in \{\forall, \exists\}$ and a quantifier free formula χ such that φ is the formula $Q_1 x_1 \dots Q_n x_n \chi$.

We will show that every \mathcal{L} -formula is provably equivalent to a formula in prenex normal form.

2.5.4. Lemma. *If φ, ψ are \mathcal{L} -formulas and $x \notin \text{Fr } \psi$, $y \notin \text{Fr } \varphi$ then*

- (i) $\vdash ((\forall x\varphi) \wedge (\forall y\psi)) \leftrightarrow \forall xy(\varphi \wedge \psi)$.
- (ii) $\vdash ((\forall x\varphi) \vee (\forall y\psi)) \leftrightarrow \forall xy(\varphi \vee \psi)$.
- (iii) $\vdash ((\exists x\varphi) \vee (\exists y\psi)) \leftrightarrow \exists xy(\varphi \vee \psi)$.
- (iv) $\vdash ((\exists x\varphi) \wedge (\exists y\psi)) \leftrightarrow \exists xy(\varphi \wedge \psi)$.

Proof. In all statements we may assume that $x \neq y$. Otherwise, our assumption $x \notin \text{Fr } \psi$ and $y \notin \text{Fr } \varphi$ implies that neither x nor y occurs freely in any of the formulas under consideration; now observe that $\vdash \varphi \leftrightarrow \forall x\varphi$ for each formula φ and every variable $x \notin \text{Fr}(\varphi)$.

(i) By the Completeness Theorem 2.4.2 it suffices to show $\models ((\forall x\varphi) \wedge (\forall y\psi)) \leftrightarrow \forall xy(\varphi \wedge \psi)$. Hence we have to take an \mathcal{L} -structure \mathcal{M} and an assignment h of \mathcal{M} and we have to show that $\mathcal{M} \models ((\forall x\varphi) \wedge (\forall y\psi))[h] \iff \mathcal{M} \models \forall xy(\varphi \wedge \psi)[h]$. The implication “ \Leftarrow ” is clear. For the implication “ \Rightarrow ” we use our assumption $x \notin \text{Fr } \psi$ and $y \notin \text{Fr } \varphi$.

(ii). By the Completeness Theorem 2.4.2 it suffices to show $\models ((\forall x\varphi) \vee (\forall y\psi)) \leftrightarrow \forall xy(\varphi \vee \psi)$. Hence we have to take an \mathcal{L} -structure \mathcal{M} and an assignment h of \mathcal{M} and we have to show that

$$\mathcal{M} \models (\forall x\varphi) \vee (\forall y\psi)[h] \iff \mathcal{M} \models \forall xy(\varphi \vee \psi)[h].$$

“ \Rightarrow ”: We may assume that $\mathcal{M} \models (\forall x\varphi)[h]$. Since y does not occur free in φ we get $\mathcal{M} \models (\forall xy\varphi)[h]$. But then $\mathcal{M} \models \forall xy(\varphi \vee \psi)[h]$ as well.

“ \Leftarrow ”: Suppose $\mathcal{M} \not\models (\forall x\varphi) \vee (\forall y\psi)[h]$. Hence there are $a, b \in |\mathcal{M}|$ with $\mathcal{M} \not\models \varphi[h(\frac{x}{a})]$ and $\mathcal{M} \not\models \psi[h(\frac{y}{b})]$. Since $x \notin \text{Fr } \psi$, $y \notin \text{Fr } \varphi$ and $x \neq y$ we obtain $\mathcal{M} \not\models \varphi \vee \psi[h(\frac{x}{a})(\frac{y}{b})]$. Hence $\mathcal{M} \not\models \forall xy(\varphi \vee \psi)[h]$.

(iii) and (iv) follow from (i) and (ii) by contraposition. \square

2.5.5. Lemma. *If φ is an \mathcal{L} -formula, $x, y \in \text{Vbl}$ and x is free in φ for y then*

$$\vdash (\forall x\varphi) \leftrightarrow (\forall y\varphi(x/y)) \text{ and } \vdash (\exists x\varphi) \leftrightarrow (\exists y\varphi(x/y)).$$

Proof. From the Completeness Theorem 2.4.2. \square

2.5.6. Prenex Normal Form Theorem

Every \mathcal{L} -formula is provably equivalent to a formula in prenex normal form.

Proof. By induction on the complexity of φ , where φ is already in prenex normal form if φ is quantifier free.

If $\varphi = \neg\psi$ or $\varphi = \forall y\psi$ and ψ is provably equivalent to a formula in prenex normal form then clearly φ also is provably equivalent to a formula in prenex normal form.

It remains to show that $\varphi \wedge \psi$ is provably equivalent to a formula in prenex normal form provided φ and ψ have this property.

So assume $\vdash \varphi \leftrightarrow Q_1x_1 \dots Q_nx_n \chi$ and $\vdash \psi \leftrightarrow P_1y_1 \dots P_ky_k \delta$ with quantifier free formulas χ, δ and $Q_i, P_j \in \{\forall, \exists\}$.

Using 2.5.5 n -times we may substitute all variables x_i in the string $Q_1x_1 \dots Q_nx_n \chi$ by variables which do not occur in $P_1y_1 \dots P_ky_k \delta$. Hence we may assume that no x_i occurs in $P_1y_1 \dots P_ky_k \delta$. Applying this again to $P_1y_1 \dots P_ky_k \delta$ we may also assume that no y_j occurs in $Q_1x_1 \dots Q_nx_n \chi$. Moreover we may assume that $k = n$, otherwise fix this by placing quantifiers in front of one of the formulas containing a new variable. Now we can apply 2.5.4(i) and (iv) to obtain

$$\vdash Q_1x_1 \dots Q_nx_n \chi \wedge P_1y_1 \dots P_ky_n \delta \leftrightarrow Q_1x_1 P_1y_1 \dots Q_nx_n P_nx_n (\chi \wedge \delta).$$

Hence also $\varphi \wedge \psi$ is provably equivalent to $Q_1x_1P_1y_1\dots Q_nx_nP_nx_n(\chi \wedge \delta)$, which is in prenex normal form. \square

2.5.7. *Example.* Let $\varphi = Q_1x_1\dots Q_nx_n \chi$ and $\varphi' = Q'_1y_1\dots Q'_ky_k \chi'$ be formulas in prenex normal form. We want to find a formula ϑ in prenex normal form such that $\vdash (\varphi \rightarrow \varphi') \leftrightarrow \vartheta$.

Let u_1, \dots, u_n be variables neither occurring in φ nor in φ' such that $u_i = u_j$ for all $i, j \in \{1, \dots, n\}$ with $x_i = x_j$. Applying *n*-times 2.5.5 we see that $\vdash \varphi \leftrightarrow Q_1u_1\dots Q_nu_n \chi(x_1/u_1)\dots(x_n/u_n)$. Hence we may replace χ by $\chi(x_1/u_1)\dots(x_n/u_n)$ and x_i by u_i if necessary and so we may assume that no x_i occurs in φ' . By applying the same argument to φ' instead of φ we may also assume that no y_j occurs in φ .

For $i \in \{1, \dots, n\}$, let $Q_i^* := \begin{cases} \forall & \text{if } Q_i \text{ is } \exists, \\ \exists & \text{if } Q_i \text{ is } \forall. \end{cases}$

Let

$$\vartheta := Q_1^*x_1\dots Q_n^*x_nQ'_1y_1\dots Q'_ky_k(\chi \rightarrow \chi').$$

By completeness, $\vdash \neg\varphi \leftrightarrow Q_1^*x_1\dots Q_n^*x_n \neg\chi$. Since $\vdash (\varphi \rightarrow \varphi') \leftrightarrow (\neg\varphi \vee \varphi')$ we obtain $\vdash (\varphi \rightarrow \varphi') \leftrightarrow \vartheta$ from the completeness theorem (keeping in mind that none of the x_i occurs in φ' and none of the y_j occurs in φ).

3. REPRESENTATION OF RECURSIVE FUNCTIONS IN ARITHMETIC

3.1. Definition.

- (i) A first order language (of predicate logic) is called **numerical**, if it contains the constant symbol 0 and the unary function symbol S . For a numerical language \mathcal{L} and $n \in \omega$ we write $S^n(x)$ for the n -fold substitution of x by $S(x)$ (so $S^0(x) = x$ and $S^{n+1}(x) = S(S^n(x))$). Further, we write

$$\underline{n} := S^n(0).$$

So \underline{n} is a constant term of \mathcal{L} (i.e. a term without variables).

Let us fix a numerical language \mathcal{L} and a set $\Sigma \subseteq \text{Sen}(\mathcal{L})$ of \mathcal{L} -sentences.

- (ii) A set $R \subseteq \omega^n$ is said to be **represented in Σ** , or **Σ -represented**, if there is some $\varphi(x_1, \dots, x_n) \in \text{Fml}(\mathcal{L})$ such that for all $a_1, \dots, a_n \in \omega$ we have:

$$\begin{aligned} R(a_1, \dots, a_n) &\Rightarrow \Sigma \vdash \varphi(\underline{a_1}, \dots, \underline{a_n}) \text{ and} \\ \neg R(a_1, \dots, a_n) &\Rightarrow \Sigma \vdash \neg \varphi(\underline{a_1}, \dots, \underline{a_n}). \end{aligned}$$

In this case we say φ **represents R in Σ** , or **R is represented by φ in Σ** . Observe that for consistent Σ ,

- both implications become equivalences,
- we can in general **not** replace the second implication by

$$R(a_1, \dots, a_n) \Leftarrow \Sigma \vdash \varphi(\underline{a_1}, \dots, \underline{a_n}).$$

Also see question 20.

- (iii) A function $F : \omega^n \rightarrow \omega$ is called **represented in Σ** or **Σ -representable**, if there is some $\varphi(x_1, \dots, x_n, y) \in \text{Fml}(\mathcal{L})$ such that for all $a_1, \dots, a_n \in \omega$ we have:

$$\Sigma \vdash \forall y \left(\varphi(\underline{a_1}, \dots, \underline{a_n}, y) \leftrightarrow y \doteq \underline{F(a_1, \dots, a_n)} \right).$$

In this case we say φ **represents F in Σ** .

- (iv) We say that an \mathcal{L} -**Term** $t(x_1, \dots, x_n)$ **represents a function $F : \omega^n \rightarrow \omega$** in Σ if for all $a_1, \dots, a_n \in \omega$ we have:

$$\Sigma \vdash t(\underline{a_1}, \dots, \underline{a_n}) \doteq \underline{F(a_1, \dots, a_n)}.$$

Observe that in this case F is represented by $t(x_1, \dots, x_n) \doteq y$ in Σ .

Remark: The intuition here is the following. If Σ and \mathcal{L} are finite and $R \subseteq \omega^n$ is represented by Σ , then there is an algorithm that decides whether $R(a_1, \dots, a_n)$ holds or not: The algorithm lists all proofs of Σ and responds 'yes' (and stops) when a proof of $\varphi(\underline{a_1}, \dots, \underline{a_n})$ appears, and responds 'no' (and stops) when a proof of $\neg \varphi(\underline{a_1}, \dots, \underline{a_n})$ appears. By the Church-Turing thesis, R should be recursive. We shall prove this in 5.2.10 and in fact this is the strongest argument supporting the Church-Turing thesis.

The goal in this section is to find a particular finite Σ in a particular finite language so that all recursive functions and relations are indeed representable by Σ .

3.2. Lemma. *Let \mathcal{L} be a numerical language and let $\Sigma \subseteq \text{Sen}(\mathcal{L})$ with $\Sigma \vdash \underline{1} \neq 0$. Then a set $R \subseteq \omega^n$ is Σ -representable if and only if its characteristic function $\mathbb{1}_R : \omega^n \rightarrow \omega$ is Σ -representable.*

Proof. First assume that $\mathbb{1}_R$ is represented by $\varphi(x_1, \dots, x_n, y)$ in Σ . Then R is represented by $\varphi(x_1, \dots, x_n, \underline{1})$ in Σ because for $a_1, \dots, a_n \in \omega$ we know that

$$\begin{aligned} & \Sigma \vdash \forall y \left(\varphi(\underline{a}_1, \dots, \underline{a}_n, y) \leftrightarrow y \doteq \underline{\mathbb{1}_R(a_1, \dots, a_n)} \right) \text{ and so} \\ (*) & \Sigma \vdash \underline{1} \doteq \underline{\mathbb{1}_R(a_1, \dots, a_n)} \rightarrow \varphi(\underline{a}_1, \dots, \underline{a}_n, \underline{1}) \text{ and} \\ (+) & \Sigma \vdash \neg \underline{1} \doteq \underline{\mathbb{1}_R(a_1, \dots, a_n)} \rightarrow \neg \varphi(\underline{a}_1, \dots, \underline{a}_n, \underline{1}). \end{aligned}$$

Hence

$$R(a_1, \dots, a_n) \Rightarrow 1 = \mathbb{1}_R(a_1, \dots, a_n) \Rightarrow \vdash \underline{1} \doteq \underline{\mathbb{1}_R(a_1, \dots, a_n)} \stackrel{\text{by } (*)}{\Rightarrow} \Sigma \vdash \varphi(\underline{a}_1, \dots, \underline{a}_n, \underline{1})$$

and

$$\begin{aligned} \neg R(a_1, \dots, a_n) \Rightarrow 0 = \mathbb{1}_R(a_1, \dots, a_n) \Rightarrow & \text{(because } \Sigma \vdash \underline{1} \neq 0) \\ \Rightarrow \Sigma \vdash \neg \underline{1} \doteq \underline{\mathbb{1}_R(a_1, \dots, a_n)} & \stackrel{\text{by } (+)}{\Rightarrow} \Sigma \vdash \neg \varphi(\underline{a}_1, \dots, \underline{a}_n, \underline{1}). \end{aligned}$$

Conversely, suppose R is represented by $\varphi(x_1, \dots, x_n)$ in Σ . Then $\mathbb{1}_R$ is represented in Σ by

$$\psi(x_1, \dots, x_n, y) := (\varphi(x_1, \dots, x_n) \wedge y \doteq \underline{1}) \vee (\neg \varphi(x_1, \dots, x_n) \wedge y \doteq \underline{0}).$$

To see this, take $a_1, \dots, a_n \in \omega$. Using the generalisation theorem (see question 19), it suffices to show

$$\Sigma \vdash \psi(\underline{a}_1, \dots, \underline{a}_n, y) \leftrightarrow y \doteq \underline{\mathbb{1}_R(a_1, \dots, a_n)}.$$

Case 1. We have $R(a_1, \dots, a_n)$.

As R is represented by φ in Σ we know $\Sigma \vdash \varphi(\underline{a}_1, \dots, \underline{a}_n)$. Using $\Sigma \vdash \underline{1} \neq 0$ this implies

$$\begin{aligned} & \Sigma \vdash \psi(\underline{a}_1, \dots, \underline{a}_n, y) \leftrightarrow y \doteq \underline{1} \text{ and so} \\ & \Sigma \vdash \psi(\underline{a}_1, \dots, \underline{a}_n, y) \leftrightarrow y \doteq \underline{\mathbb{1}_R(a_1, \dots, a_n)} \end{aligned}$$

Case 2. We have $\neg R(a_1, \dots, a_n)$.

As R is represented by φ in Σ we know $\Sigma \vdash \neg \varphi(\underline{a}_1, \dots, \underline{a}_n)$. Using $\Sigma \vdash \underline{1} \neq 0$ this implies

$$\begin{aligned} & \Sigma \vdash \psi(\underline{a}_1, \dots, \underline{a}_n, y) \leftrightarrow y \doteq \underline{0} \text{ and so} \\ & \Sigma \vdash \psi(\underline{a}_1, \dots, \underline{a}_n, y) \leftrightarrow y \doteq \underline{\mathbb{1}_R(a_1, \dots, a_n)} \end{aligned}$$

□

As announced already, we shall now define a set of sentences that represents all recursive functions. Let $\mathcal{L}_{\bar{\omega}}$ be the language

$$\mathcal{L}_{\bar{\omega}} := \{+, \cdot, S, <, 0\},$$

where $+$ and \cdot are binary function symbols, S is a unary function symbol, $<$ is a binary relation symbol and 0 is a constant symbol.

Further, let $\bar{\omega}$ be the structure $(\omega, +, \cdot, S, <, 0)$ with the natural interpretation of the non-logical $\mathcal{L}_{\bar{\omega}}$ -symbols; where $S(n) = n + 1$ is the successor function. $\bar{\omega}$ is called the **standard model**.

Formally, we should make a notational difference between the operations in $\bar{\omega}$ and the symbols of $\mathcal{L}_{\bar{\omega}}$ denoting them. However, for better readability and in order to avoid cumbersome formulas we won't do that.

Let Ω denote the following set of nine $\mathcal{L}_{\bar{\omega}}$ -sentences (where we also write Sx instead of $S(x)$):

- | | | |
|-------------------------------|--|---|
| $\Omega 1.$ | $\forall x Sx \neq 0$ | 0 is not a successor |
| $\Omega 2.$ | $\forall x \forall y Sx \doteq Sy \rightarrow x \doteq y$ | The successor function is injective |
| $\Omega 3.$ | $\forall x x + 0 \doteq x$ | 0 is a right identity w.r.t. $+$ |
| $\Omega 4.$ | $\forall x \forall y x + Sy \doteq S(x + y)$ | recursive definition of $+$
from the successor function |
| $\Omega 5.$ | $\forall x x \cdot 0 \doteq 0$ | |
| $\Omega 6.$ | $\forall x \forall y x \cdot Sy \doteq x \cdot y + x$ | recursive definition of \cdot from $+$ |
| $\Omega 7.$ | $\forall x \neg x < 0$ | |
| $\Omega 8.$ | $\forall x \forall y x < Sy \leftrightarrow x < y \vee x \doteq y$ | recursive definition of the
successor function using $<$ |
| $\Omega 9.$ | $\forall x \forall y x < y \vee x \doteq y \vee y < x$ | Totality of $<$ |

The set Ω is called **Robinson Arithmetic**. Obviously the standard model $\bar{\omega}$ is a model of Ω . On examples sheet 5 we will see some 'non-standard' models of Ω .

In order to verify whether a given $\mathcal{L}_{\bar{\omega}}$ -formula φ logically follows from Ω we need to check that every model of Ω is a model of φ . Since there are bizarre models of Ω it is hopeless to obtain a good understanding of these models. However, we will now see that it is correct to think of each model as having the standard model as a kind of initial building block (see 3.3 below).

First recall that a **homomorphism** $\mathcal{M} \rightarrow \mathcal{N}$ between \mathcal{L} -structures for an arbitrary language \mathcal{L} is a map $f : |\mathcal{M}| \rightarrow |\mathcal{N}|$ between the universes of the structures that respects all non-logical symbols, thus

- (i) For every relation symbol R of \mathcal{L} of arity n and all $a_1, \dots, a_n \in |\mathcal{M}|$ we have

$$\mathcal{M} \models R[a_1, \dots, a_n] \Rightarrow \mathcal{N} \models R[f(a_1), \dots, f(a_n)]$$

- (ii) For every function symbol F of \mathcal{L} of arity n and all $a_1, \dots, a_n \in |\mathcal{M}|$ we have

$$f(F^{\mathcal{M}}(a_1, \dots, a_n)) = F^{\mathcal{N}}(f(a_1), \dots, f(a_n))$$

- (iii) For every constant symbol c of \mathcal{L} we have

$$f(c^{\mathcal{M}}) = c^{\mathcal{N}}.$$

A compact way to say the same thing is: f preserves all atomic \mathcal{L} -formulas, i.e. If $\varphi(x_1, \dots, x_n)$ is an atomic \mathcal{L} -formula and $a_1, \dots, a_n \in |\mathcal{M}|$ then

$$\mathcal{M} \models \varphi[a_1, \dots, a_n] \Rightarrow \mathcal{N} \models \varphi[f(a_1), \dots, f(a_n)].$$

Also, recall that an **embedding** $\mathcal{M} \rightarrow \mathcal{N}$ between \mathcal{L} -structures is a map $f : |\mathcal{M}| \rightarrow |\mathcal{N}|$ which respect all quantifier free formulas. Explicitly, this is equivalent to saying that f is an injective homomorphism $\mathcal{M} \rightarrow \mathcal{N}$ (where injective just means injective as a map) such that in condition (i) of the definition of a homomorphism above, we have equivalence.

3.3. Proposition. *If \mathcal{M} is a model of Ω then there is a unique $\mathcal{L}_{\bar{\omega}}$ -homomorphism $\varepsilon : \bar{\omega} \rightarrow \mathcal{M}$, given by*

$$\varepsilon(n) = \underline{n}^{\mathcal{M}}.$$

ε is an embedding and has the following properties:

- (i) If $a \in |\mathcal{M}|$ and $n \in \omega$ with $a <^{\mathcal{M}} \varepsilon(n)$, then there is some $m \in \omega$, $m < n$ with $\varepsilon(m) = a$.
- (ii) If $a \in |\mathcal{M}|$ is not in the image of ε , then $\varepsilon(n) <^{\mathcal{M}} a$ for all $n \in \omega$.

Proof. Since n is the interpretation of $S^n(0)$ in $\bar{\omega}$ and ε is supposed to respect S , we must define ε via $\varepsilon(n) := (S^n(0))^{\mathcal{M}} = \underline{n}^{\mathcal{M}}$ (which implies uniqueness). We first show that ε is an embedding.

- ε respects S because

$$\varepsilon(S^{\bar{\omega}}n) = \varepsilon(n+1) = (S^{n+1}0)^{\mathcal{M}} = S^{\mathcal{M}}\varepsilon(n).$$

- ε is injective:

Otherwise there are $n < m$ in ω with $(S^n 0)^{\mathcal{M}} = (S^m 0)^{\mathcal{M}}$. Then, by induction, using $\Omega 2$ we see that $0^{\mathcal{M}} = (S^{m-n} 0)^{\mathcal{M}}$ and this contradicts $m - n > 0$ and $\Omega 1$.

- $\varepsilon(n+m) = \varepsilon(n) +^{\mathcal{M}} \varepsilon(m)$:

If $m = 0$, this holds true by $\Omega 3$. Inductively we see that

$$\begin{aligned} \varepsilon(n+m+1) &= \varepsilon(S(n+m)) \\ &= S^{\mathcal{M}}(\varepsilon(n+m)) && \text{(since } \varepsilon \text{ respects } S) \\ &= S^{\mathcal{M}}(\varepsilon(n) +^{\mathcal{M}} \varepsilon(m)) && \text{(by induction)} \\ &= \varepsilon(n) +^{\mathcal{M}} S^{\mathcal{M}}(\varepsilon(m)) = \varepsilon(n) +^{\mathcal{M}} \varepsilon(m+1). && \text{(by } \Omega 4) \end{aligned}$$

- $\varepsilon(n \cdot m) = \varepsilon(n) \cdot^{\mathcal{M}} \varepsilon(m)$:

If $m = 0$, this holds true by $\Omega 5$. Inductively we see that

$$\begin{aligned} \varepsilon(n \cdot (m+1)) &= \varepsilon(n \cdot m + n) \\ &= \varepsilon(n \cdot m) +^{\mathcal{M}} \varepsilon(n) && \text{(since } \varepsilon \text{ respects } +) \\ &= \varepsilon(n) \cdot^{\mathcal{M}} \varepsilon(m) +^{\mathcal{M}} \varepsilon(n) && \text{(by induction)} \\ &= \varepsilon(n) \cdot^{\mathcal{M}} S^{\mathcal{M}}\varepsilon(m) && \text{(by } \Omega 6) \\ &= \varepsilon(n) \cdot^{\mathcal{M}} \varepsilon(m+1). \end{aligned}$$

- $n < m \iff \varepsilon(n) <^{\mathcal{M}} \varepsilon(m)$:

If $m = 0$, this holds true by **$\Omega 7$** . By induction on m we get

$$\begin{aligned}
n < m + 1 &\iff n < m \text{ or } n = m \\
&\iff \varepsilon(n) <^{\mathcal{M}} \varepsilon(m) \text{ or } \varepsilon(n) = \varepsilon(m) && \text{(by induction} \\
&&& \text{and as } \varepsilon \text{ is injective)} \\
&\iff \varepsilon(n) <^{\mathcal{M}} S^{\mathcal{M}} \varepsilon(m) && \text{(by } \mathbf{\Omega 8}) \\
&\iff \varepsilon(n) <^{\mathcal{M}} \varepsilon(m + 1).
\end{aligned}$$

So we have shown that ε is an embedding $\bar{\omega} \hookrightarrow \mathcal{M}$.

(i). For $a \in |\mathcal{M}|$ and $n \in \omega$ with $a <^{\mathcal{M}} \varepsilon(n)$ we need to find some $m \in \omega$, $m < n$ with $\varepsilon(m) = a$.

By induction on n : If $n = 0$, so then $\varepsilon(n) = 0^{\mathcal{M}}$ and by **$\Omega 7$** there is no $a \in |\mathcal{M}|$ with $a <^{\mathcal{M}} \varepsilon(n)$.

Now assume $a \in |\mathcal{M}|$ with $a <^{\mathcal{M}} \varepsilon(n + 1) = S^{\mathcal{M}} \varepsilon(n)$. From the implication \rightarrow in **$\Omega 8$** we know $a <^{\mathcal{M}} \varepsilon(n)$ or $a = \varepsilon(n)$. In the first case we may find $m < n + 1$ with $\varepsilon(m) = a$ using induction; in the second case we take $m = n$.

(ii) Take $a \in |\mathcal{M}|$ not in the image of ε and let $n \in \omega$. We need to show $\varepsilon(n) <^{\mathcal{M}} a$. Otherwise, using **$\Omega 9$** we know $\varepsilon(n) = a$, which is not possible by choice of a , or, $a <^{\mathcal{M}} \varepsilon(n)$, which contradicts (i) and the choice of a . \square

Here is a first application of **3.3**:

3.4. Corollary. *For every $n \in \omega$ we have*

$$\mathbf{\Omega} \vdash \left(x < \underline{n+1} \leftrightarrow (x \doteq 0 \vee \dots \vee x \doteq \underline{n}) \right).$$

Proof. For every model \mathcal{M} of **Ω** and all $a \in |\mathcal{M}|$ we know

$$a < \underline{n+1}^{\mathcal{M}} \iff (a = 0^{\mathcal{M}} \text{ or } \dots \text{ or } a = \underline{n}^{\mathcal{M}})$$

by **3.3**. Hence the result follows. \square

Of course **3.4** can also be deduced directly from **$\Omega 8$** und **$\Omega 7$** by induction on n .

3.5. Representability Theorem

*For every language \mathcal{L} extending the language $\mathcal{L}_{\bar{\omega}}$, every recursive function $\omega^n \rightarrow \omega$ and every recursive relation $\subseteq \omega^n$ ($n \in \omega$) is represented in all sets of \mathcal{L} -formulas Σ which prove **Ω** (so $\Sigma \vdash \mathbf{\Omega}$).*

*(In **5.2.10** we will prove the converse of the Representability Theorem.)*

Proof. The theorem follows from the case $\Sigma = \mathbf{\Omega}$ and $\mathcal{L} = \mathcal{L}_{\bar{\omega}}$ using standard arguments from predicate logic (make sure you can do this).

By **3.2** it suffices to show that every recursive function is represented in **Ω** . In order to do so, we show that the elementary functions from **R1** of definition **1.1.1** are represented in **Ω** and that the rules **R2** and **R3** of **1.1.1**, when fed with functions that are represented in **Ω** , return functions that are again represented in **Ω** .

For later use we shall also keep track of the quantifiers used in the representing formulas. Note (by observation of the proof of **3.2**) that the translations of the representation of a relation into the representation of the graph of its characteristic function and vice versa, do not introduce any quantifiers.

3.5.1. Every elementary function from **R1** of definition 1.1.1 is represented in Ω by a quantifier free $\mathcal{L}_{\bar{\omega}}$ -formula.

Proof of 3.5.1.

- (a) The formula $x_1 \doteq x_2$ represents the diagonal of ω^2 in Ω because for $a = b$ we obviously have $\vdash \underline{a} \doteq \underline{b}$; further, if $a \neq b$, then $\Omega \vdash \neg \underline{a} \doteq \underline{b}$ because $\neg \underline{a} \doteq \underline{b}$ holds in every model of Ω by 3.3.
- (b) The term $x_1 + x_2$ represents addition of $\bar{\omega}$ in Ω , because $\underline{a} + \underline{b} \doteq \underline{a + b}$ holds in every model of Ω (cf. 3.3) and so $\Omega \vdash \underline{a} + \underline{b} \doteq \underline{a + b}$.
- (c) The term $x_1 \cdot x_2$ represents multiplication of $\bar{\omega}$ in Ω , because $\underline{a} \cdot \underline{b} \doteq \underline{a \cdot b}$ holds in every model of Ω (cf. 3.3) and so $\Omega \vdash \underline{a} \cdot \underline{b} \doteq \underline{a \cdot b}$.
- (d) The formula $x_1 < x_2$ represents the relation $\{(a, b) \in \omega^2 \mid a < b\}$ in Ω : If $a < b$, then $\underline{a} < \underline{b}$ holds in every model of Ω by 3.3, thus $\Omega \vdash \underline{a} < \underline{b}$. Further, if $\neg a < b$, then by 3.3, $\neg \underline{a} < \underline{b}$ holds in every model of Ω (here we use that the map ε in 3.3 is not only an injective homomorphism, but also an embedding), thus $\Omega \vdash \neg \underline{a} < \underline{b}$.
- (e) The function $\mathbf{1}_{\leq}$ is represented in Ω by a quantifier free formula: By (a) and (d) the relation \leq of $\bar{\omega}$ is represented in Ω by the formula $x_1 < x_2 \vee x_1 \doteq x_2$ (now we may use 3.2 again, which does not introduce quantifiers).
- (f) Obviously, the term x_i represents the coordinate function I_i^n in Ω .

□

3.5.2. Let $F : \omega^n \rightarrow \omega$ and $G_1, \dots, G_n : \omega^k \rightarrow \omega$ be represented in Ω by the formulas

$$\varphi(x_1, \dots, x_n, y), \quad \psi_i(x_1, \dots, x_k, y) \quad (1 \leq i \leq n),$$

respectively. Then the composition $F(G_1, \dots, G_n) : \omega^k \rightarrow \omega$ is represented in Ω by

$$\forall y_1 \dots y_n \left(\bigwedge_{i=1}^n \psi_i(x_1, \dots, x_k, y_i) \rightarrow \varphi(y_1, \dots, y_n, z) \right) \text{ and by}$$

$$\exists y_1 \dots y_n \left(\bigwedge_{i=1}^n \psi_i(x_1, \dots, x_k, y_i) \wedge \varphi(y_1, \dots, y_n, z) \right)$$

Proof of 3.5.2. Let us write

$$\gamma(x_1, \dots, x_k, z) := \forall y_1 \dots y_n \left(\bigwedge_{i=1}^n \psi_i(x_1, \dots, x_k, y_i) \rightarrow \varphi(y_1, \dots, y_n, z) \right) \text{ and}$$

$$\delta(x_1, \dots, x_k, z) := \exists y_1 \dots y_n \left(\bigwedge_{i=1}^n \psi_i(x_1, \dots, x_k, y_i) \wedge \varphi(y_1, \dots, y_n, z) \right)$$

Pick $a_1, \dots, a_k \in \omega$ and write $a = (a_1, \dots, a_k)$. We have to show

$$\Omega \vdash \gamma(\underline{a_1}, \dots, \underline{a_k}, z) \leftrightarrow z \doteq \underline{F(G_1(a), \dots, G_n(a))} \text{ and}$$

$$\Omega \vdash \delta(\underline{a_1}, \dots, \underline{a_k}, z) \leftrightarrow z \doteq \underline{F(G_1(a), \dots, G_n(a))}.$$

It suffices to show that for every model \mathcal{M} of Ω and each element τ of the universe of \mathcal{M} the following are equivalent:

- (a) $\mathcal{M} \models \gamma(\underline{a_1}, \dots, \underline{a_k}, z)[\tau]$,

(b) $\mathcal{M} \models \delta(\underline{a}_1, \dots, \underline{a}_k, z)[\tau]$ and

(c) $\tau = \underline{F(G_1(a), \dots, G_n(a))}^{\mathcal{M}}$.

We write $b_i = G_i(a)$ ($1 \leq i \leq n$). By assumption we have

(*) $\Omega \vdash \psi_i(\underline{a}_1, \dots, \underline{a}_k, y_i) \leftrightarrow y_i \doteq \underline{G_i(a)}$,

and

(†) $\Omega \vdash \varphi(\underline{b}_1, \dots, \underline{b}_n, z) \leftrightarrow z \doteq \underline{F(b_1, \dots, b_n)}$.

(a) \Rightarrow (b).

We interpret y_i as $\underline{b_i}^{\mathcal{M}}$ ($1 \leq i \leq n$) and show that

- $\mathcal{M} \models \psi_i(\underline{a}_1, \dots, \underline{a}_k, \underline{b_i}^{\mathcal{M}})$ for each $i \in \{1, \dots, n\}$ and
- $\mathcal{M} \models \varphi(\underline{b}_1, \dots, \underline{b}_n, z)[\tau]$.

The first item holds, because of $\mathcal{M} \models \Omega$ and because of the implication \leftarrow in (*). By assumption (a) and the choice of γ we therefore see $\mathcal{M} \models \varphi(\underline{b}_1, \dots, \underline{b}_n, z)[\tau]$, as required.

(b) \Rightarrow (c).

By (b) there are $\tau_1, \dots, \tau_n \in |\mathcal{M}|$ with

$$\mathcal{M} \models \bigwedge_{i=1}^n \psi_i[\underline{a}_1^{\mathcal{M}}, \dots, \underline{a}_k^{\mathcal{M}}, \tau_i] \text{ and } \mathcal{M} \models \varphi[\tau_1, \dots, \tau_n, \tau].$$

From the implication \rightarrow in (*) and $\mathcal{M} \models \Omega$ we get $\tau_i = \underline{G_i(a)}^{\mathcal{M}} = \underline{b_i}^{\mathcal{M}}$ and so $\mathcal{M} \models \varphi(\underline{b}_1, \dots, \underline{b}_n, z)[\tau]$. From the implication \rightarrow in (†) we get $\tau = \underline{F(b_1, \dots, b_n)}^{\mathcal{M}} = \underline{F(G_1(a), \dots, G_n(a))}^{\mathcal{M}}$, as required.

(c) \Rightarrow (a).

Take any $\tau_1, \dots, \tau_n \in |\mathcal{M}|$ with

$$\mathcal{M} \models \bigwedge_{i=1}^n \psi_i[\underline{a}_1^{\mathcal{M}}, \dots, \underline{a}_k^{\mathcal{M}}, \tau_i]$$

From the implication \rightarrow in (*) and $\mathcal{M} \models \Omega$ we get $\tau_i = \underline{G_i(a)}^{\mathcal{M}} = \underline{b_i}^{\mathcal{M}}$ again. As $\tau = \underline{F(G_1(a), \dots, G_n(a))}^{\mathcal{M}} = \underline{F(b_1, \dots, b_n)}^{\mathcal{M}}$ by assumption (c), the implication \leftarrow in (†) together with $\mathcal{M} \models \Omega$ give $\mathcal{M} \models \varphi(\underline{b}_1, \dots, \underline{b}_n, z)[\tau]$. This means $\mathcal{M} \models \varphi[\tau_1, \dots, \tau_n, \tau]$ as required.

This finishes the proof of 3.5.2. \square

3.5.3. Let $F : \omega^n \times \omega \rightarrow \omega$ be represented by $\varphi(x_1, \dots, x_n, u, y)$ in Ω such that for each $a \in \omega^n$ there is some $b \in \omega$ with $F(a, b) = 0$.

Let $G : \omega^n \rightarrow \omega$ be defined by

$$G(a) := \mu x (F(a, x) = 0).$$

Then G is represented in Ω by the formula

$$\varphi(x_1, \dots, x_n, z, 0) \wedge \forall u \left(u < z \rightarrow \neg \varphi(x_1, \dots, x_n, u, 0) \right).$$

Proof of 3.5.3. Let us write

$$\psi(x_1, \dots, x_n, z) := \varphi(x_1, \dots, x_n, z, 0) \wedge \forall u \, u < z \rightarrow \neg \varphi(x_1, \dots, x_n, u, 0)$$

For $a_1, \dots, a_n \in \omega$ we have to show

$$\Omega \vdash \psi(\underline{a_1}, \dots, \underline{a_n}, z) \leftrightarrow z \doteq \underline{G(a_1, \dots, a_n)}.$$

It suffices to show for every model \mathcal{M} of Ω and all $\tau \in |\mathcal{M}|$ that

$$(*) \quad \mathcal{M} \models \psi(\underline{a_1}, \dots, \underline{a_n}, z)[\tau] \iff \tau = \underline{G(a_1, \dots, a_n)}^{\mathcal{M}}.$$

By assumption, we know for all $d \in \omega$:

$$(+)$$

$$\Omega \vdash \varphi(\underline{a_1}, \dots, \underline{a_n}, \underline{d}, y) \leftrightarrow y \doteq \underline{F(a_1, \dots, a_n, d)}$$

Let $b = G(a_1, \dots, a_n)$. By definition of G , b is the smallest zero of $F(a, x)$ and from the implication \leftarrow in (+) we know

$$(\dagger) \quad \mathcal{M} \models \varphi(\underline{a_1}, \dots, \underline{a_n}, \underline{b}, 0).$$

Now we prove the equivalence (*).

\Leftarrow : So here $\tau = \underline{b}^{\mathcal{M}}$ and by (\dagger) we have $\mathcal{M} \models \varphi(\underline{a_1}, \dots, \underline{a_n}, z, 0)[\tau]$. It remains to show that for every $\rho \in |\mathcal{M}|$ with $\rho <^{\mathcal{M}} \tau$ we have $\mathcal{M} \models \neg \varphi(\underline{a_1}, \dots, \underline{a_n}, u, 0)[\rho]$.

As $\tau = \underline{b}^{\mathcal{M}}$ and $\mathcal{M} \models \Omega$, we may apply 3.4 and get some $c \in \omega$, $c < b$ with $\rho = \underline{c}^{\mathcal{M}}$. By choice of b , we know $F(a, c) \neq 0$. From the implication \rightarrow in (+) (by setting $y = 0$ and choosing $d = c$) we then get $\Omega \vdash \neg \varphi(\underline{a_1}, \dots, \underline{a_n}, \underline{c}, 0)$.

Consequently, $\mathcal{M} \models \neg \varphi(\underline{a_1}, \dots, \underline{a_n}, u, 0)[\rho]$ as required.

\Rightarrow : Now suppose $\mathcal{M} \models \psi(\underline{a_1}, \dots, \underline{a_n}, z)[\tau]$. Then $\mathcal{M} \models \varphi(\underline{a_1}, \dots, \underline{a_n}, z, 0)[\tau]$ and by (\dagger) we cannot have $\underline{b}^{\mathcal{M}} <^{\mathcal{M}} \tau$.

So in order to confirm $\tau = \underline{G(a_1, \dots, a_n)}^{\mathcal{M}} = \underline{b}^{\mathcal{M}}$ we may use $\Omega 9$ and show that $\tau <^{\mathcal{M}} \underline{b}^{\mathcal{M}}$ does not hold either:

Suppose we have $\tau <^{\mathcal{M}} \underline{b}^{\mathcal{M}}$. Then by 3.4, there is some $c \in \omega$, $c < b$ with $\tau = \underline{c}^{\mathcal{M}}$. From the implication \rightarrow in (+) (with $c = d$ and $y = 0$) and $\mathcal{M} \models \varphi(\underline{a_1}, \dots, \underline{a_n}, \underline{c}, 0)$ we then obtain $F(a, c) = 0$, which contradicts the minimality of b .

This finishes the proof of 3.5.3. \square

Using 3.5.1, 3.5.2 and 3.5.3, it is now clear that every recursive function is represented in Ω , which finishes the proof of 3.5. \square

4. ARITHMETISATION OF LOGIC: GÖDELISATION

In this section we fix a *finite or countable* language

$$\mathcal{L} = (\lambda : \mathcal{R} \rightarrow \mathbb{N}, \mu : \mathcal{F} \rightarrow \mathbb{N}, \mathcal{C}).$$

Recall that the cardinality of a language is the cardinality of the union $\mathcal{R} \cup \mathcal{F} \cup \mathcal{C}$ of its non-logical symbols. Also, recall that the alphabet of \mathcal{L} is the set of all logical and non-logical symbols.

Further, we assume that \mathcal{L} is given *recursively*. This means the following: We are given an *injective* map

$$[\cdot] : \text{Alphabet of } \mathcal{L} \rightarrow \omega$$

such that:

- (0) $[v_i] = 2i$ for $i \in \omega$, in particular $[\text{Vbl}] = \{2i \mid i \in \omega\}$.
- (1) The sets $[\mathcal{F}] = \{[F] \mid F \in \mathcal{F}\}$, $[\mathcal{R}] = \{[R] \mid R \in \mathcal{R}\}$ and $[\mathcal{C}] = \{[c] \mid c \in \mathcal{C}\}$ are recursive.
- (2) The maps $[\lambda] : [\mathcal{R}] \rightarrow \omega$, $[\mu] : [\mathcal{F}] \rightarrow \omega$ defined by $[\lambda]([R]) = \lambda(R)$ and $[\mu]([F]) = \mu(F)$ are recursive (i.e., have recursive graphs) and have recursive image.

If l is a letter of the alphabet of \mathcal{L} , then $[l]$ is called the **symbol number** of l .

Intuitively one should think of \mathcal{R} , \mathcal{F} and \mathcal{C} as being equal to $[\mathcal{R}]$, $[\mathcal{F}]$ and $[\mathcal{C}]$. If \mathcal{L} is finite (which is a major case later on), then conditions (1) and (2) are always satisfied, because all finite sets are in fact primitive recursive. Also, there is nothing special about the symbol numbers assigned to the variables, as long as they form a recursive subset of ω .

4.1. Definition. The **Gödel number** $\ulcorner t \urcorner$ (or just **code**) of an \mathcal{L} -term t is defined by induction as the following sequence number:

$$\ulcorner t \urcorner := \begin{cases} \ulcorner [t] \urcorner & \text{if } t \text{ is a variable or a constant symbol} \\ \ulcorner [F], \ulcorner t_1 \urcorner, \dots, \ulcorner t_n \urcorner \urcorner & \text{if } t = F(t_1, \dots, t_n) \text{ (and so } \mu(F) = n). \end{cases}$$

Notice that this indeed is well defined by the unique readability theorem for terms, see 2.1.5.

Hence we have

$$(\ulcorner F(t_1, \dots, t_n) \urcorner)_{i+1} = \ulcorner t_i \urcorner \quad (1 \leq i \leq n).$$

The **Gödel number** $\ulcorner \varphi \urcorner$ (or just **code**) of an \mathcal{L} -formula φ is defined inductively as the following sequence number:

$$\ulcorner \varphi \urcorner := \begin{cases} \ulcorner [\dot{=}], \ulcorner t_1 \urcorner, \ulcorner t_2 \urcorner \urcorner & \text{if } \varphi \text{ is of the form } t_1 \dot{=} t_2 \\ \ulcorner [R], \ulcorner t_1 \urcorner, \dots, \ulcorner t_n \urcorner \urcorner & \text{if } \varphi \text{ is of the form } R(t_1, \dots, t_n) \\ \ulcorner [\neg], \ulcorner \psi \urcorner \urcorner & \text{if } \varphi \text{ is of the form } \neg\psi \\ \ulcorner [\rightarrow], \ulcorner \varphi_1 \urcorner, \ulcorner \varphi_2 \urcorner \urcorner & \text{if } \varphi \text{ is of the form } \varphi_1 \rightarrow \varphi_2 \\ \ulcorner [\forall], \ulcorner x \urcorner, \ulcorner \psi \urcorner \urcorner & \text{if } \varphi \text{ is of the form } \forall x \psi \end{cases}$$

Again, this is well defined by the unique readability theorem for formulas, see 2.1.9. For any set S of \mathcal{L} -terms and \mathcal{L} -formulas we write

$$\ulcorner S \urcorner := \{\ulcorner s \urcorner \mid s \in S\}.$$

It should be noted that brackets and commas (which are present in our version of predicate logic) are not coded above: We don't need to do that, because we can uniquely reconstruct terms and formulas from their Gödel number: By a trivial induction on the complexity of terms and formulas we can see directly from the definition of $\ulcorner t \urcorner$ and $\ulcorner \varphi \urcorner$ that the map $s \mapsto \ulcorner s \urcorner$ (defined where $\ulcorner \cdot \urcorner$ is defined) is injective. Less trivial is the fact that the image is recursive. This is tackled next.

In what follows, remember the following properties of sequence numbers:

- (i) (See 1.4.2(ii)) For each $n \in \omega$ and all $a_0, \dots, a_n \in \omega$ we have

$$\begin{aligned} \langle \prec a_1, \dots, a_n \succ \rangle_i &= a_i < \prec a_1, \dots, a_n \succ \text{ for } 1 \leq i \leq n \text{ and} \\ \langle \prec a_0, \dots, a_{n-1} \succ \rangle_{i+1} &= a_i < \prec a_0, \dots, a_{n-1} \succ \text{ for } i < n. \end{aligned}$$

- (ii) The function $\langle \prec x_1, \dots, x_n \succ \rangle: \omega^n \rightarrow \omega$ is recursive (see 1.4.1).
 (iii) If $R \subseteq \omega^n$ is recursive, then also the set

$$\langle \prec R \succ \rangle := \{\langle \prec a_1, \dots, a_n \succ \mid R(a_1, \dots, a_n)\}$$

is recursive, because $x \in \langle \prec R \succ \rangle \iff \mathbf{Seq}(x) \wedge \ell(x) = n \wedge R((x)_1, \dots, (x)_n)$ and $\mathbf{Seq}(x)$ is recursive by 1.4.2(v).

For example, by (iii) we know that $\ulcorner \mathcal{C} \urcorner = \langle \prec \mathcal{C} \succ \rangle$ and $\ulcorner \forall \text{bl} \urcorner = \langle \prec 2\omega \succ \rangle$ are recursive.

4.2. Lemma. *The following subsets of ω are recursive: $\ulcorner \text{tm}(\mathcal{L}) \urcorner$, $\ulcorner \text{at-Fml}(\mathcal{L}) \urcorner$ (recall that $\text{at-Fml}(\mathcal{L})$ is the set of atomic \mathcal{L} -formulas) and $\ulcorner \text{Fml}(\mathcal{L}) \urcorner$.*

Proof. We already know that $\ulcorner \forall \text{bl} \urcorner$ and $\ulcorner \mathcal{C} \urcorner$ are recursive.

$\ulcorner \text{tm}(\mathcal{L}) \urcorner$ is recursive:

To see this we first formalize the inductive definition of $\ulcorner t \urcorner$, for \mathcal{L} -terms t into

$$\begin{aligned} x \in \ulcorner \text{tm}(\mathcal{L}) \urcorner &\iff x \in \ulcorner \forall \text{bl} \urcorner \vee x \in \ulcorner \mathcal{C} \urcorner \vee \left(\mathbf{Seq}(x) \wedge (x)_1 \in [\mathcal{F}] \wedge \right. \\ &\quad \text{there is some } n < x \text{ with } [\mu]((x)_1) = n \wedge \ell(x) = n + 1, \\ &\quad \left. \text{such that for all } 1 \leq i \leq n \text{ we have } (x)_{i+1} \in \ulcorner \text{tm}(\mathcal{L}) \urcorner \right) \end{aligned}$$

The formulation "there is some $n < x$ with ..." is chosen to make the statement more readable and keep it closer to the definition of terms and their Gödel numbers. We could have also said

$$\begin{aligned} x \in \ulcorner \text{tm}(\mathcal{L}) \urcorner &\iff x \in \ulcorner \forall \text{bl} \urcorner \vee x \in \ulcorner \mathcal{C} \urcorner \vee \left(\mathbf{Seq}(x) \wedge (x)_1 \in [\mathcal{F}] \wedge \right. \\ &\quad [\mu]((x)_1) = \ell(x) \dot{-} 1 \wedge \\ &\quad \left. \text{for all } 1 \leq i \leq \ell(x) \dot{-} 1 \text{ we have } (x)_{i+1} \in \ulcorner \text{tm}(\mathcal{L}) \urcorner \right) \end{aligned}$$

We stick with the first equivalence and may now use 1.4.5 to show that $\ulcorner \text{tm}(\mathcal{L}) \urcorner$

is recursive, by applying it with $F(x) = \mathbb{1}_{\ulcorner \text{tm}(\mathcal{L}) \urcorner}(x)$ and

$$G(x, v) = \begin{cases} 1 & \text{if } x \in \ulcorner \text{Vbl} \urcorner \vee x \in \ulcorner \mathcal{C} \urcorner \vee \left(\mathbf{Seq}(x) \wedge (x)_1 \in [\mathcal{F}] \wedge \right. \\ & \quad \exists n < x : ([\mu]((x)_1) = n \wedge \ell(x) = n + 1 \wedge \\ & \quad \quad \left. \forall 1 \leq i \leq n : (v)_{(x)_{i+1}+1} = 1) \right) \\ 0 & \text{otherwise.} \end{cases}$$

From the results in section 1 we know that G is indeed recursive and by 1.4.5 it remains to show that

$$(*) \quad \mathbb{1}_{\ulcorner \text{tm}(\mathcal{L}) \urcorner}(x) = G(x, \overline{\mathbb{1}_{\ulcorner \text{tm}(\mathcal{L}) \urcorner}(x)}) = G(x, \prec \mathbb{1}_{\ulcorner \text{tm}(\mathcal{L}) \urcorner}(0), \dots, \mathbb{1}_{\ulcorner \text{tm}(\mathcal{L}) \urcorner}(x-1) \succ).$$

Take a term $t = F(t_1, \dots, t_n)$ and let $x = \ulcorner t \urcorner = \prec [F], \ulcorner t_1 \urcorner, \dots, \ulcorner t_n \urcorner \succ$. Then $n < x$ and $\ulcorner t_i \urcorner = (x)_{i+1} < x$ for all $1 \leq i \leq n$. Thus

$$\mathbb{1}_{\ulcorner \text{tm}(\mathcal{L}) \urcorner}((x)_{i+1}) = 1 \text{ and } (x)_{i+1} < x.$$

Now we set $v = \overline{\mathbb{1}_{\ulcorner \text{tm}(\mathcal{L}) \urcorner}(x)} = \prec \mathbb{1}_{\ulcorner \text{tm}(\mathcal{L}) \urcorner}(0), \dots, \mathbb{1}_{\ulcorner \text{tm}(\mathcal{L}) \urcorner}(x-1) \succ$.

Since $\mathbb{1}_{\ulcorner \text{tm}(\mathcal{L}) \urcorner}((x)_{i+1})$ occurs at the $(x)_{i+1} + 1^{\text{th}}$ position in

$$(\mathbb{1}_{\ulcorner \text{tm}(\mathcal{L}) \urcorner}(0), \dots, \mathbb{1}_{\ulcorner \text{tm}(\mathcal{L}) \urcorner}(x-1)),$$

we see that

$$(v)_{(x)_{i+1}+1} = \mathbb{1}_{\ulcorner \text{tm}(\mathcal{L}) \urcorner}((x)_{i+1}) = 1.$$

This shows $G(x, \overline{\mathbb{1}_{\ulcorner \text{tm}(\mathcal{L}) \urcorner}(x)}) = G(x, v) = 1$. Conversely, if x is not the Gödel number of a term, the considerations above show that $G(x, \overline{\mathbb{1}_{\ulcorner \text{tm}(\mathcal{L}) \urcorner}(x)}) = 0$.

$\ulcorner \text{at-Fml}(\mathcal{L}) \urcorner$ is recursive:

The set

$$A := \{ \ulcorner R(t_1, \dots, t_n) \urcorner \mid n \in \mathbb{N}, t_1, \dots, t_n \in \text{tm}(\mathcal{L}), R \in \mathcal{R}, \lambda(R) = n \}$$

is recursive, because

$$\begin{aligned} x \in A &\iff \mathbf{Seq}(x) \wedge (x)_1 \in [\mathcal{R}] \wedge \\ &\quad \exists n < x : \left([\lambda]((x)_1) = n, \ell(x) = n + 1 \wedge \right. \\ &\quad \quad \left. \forall 1 \leq i \leq n : (x)_{i+1} \in \ulcorner \text{tm}(\mathcal{L}) \urcorner \right). \end{aligned}$$

and the conditions on the right hand side of this equivalence are all recursive.

The set

$$B = \{ \ulcorner t_1 \doteq t_2 \urcorner \mid t_1, t_2 \in \text{tm}(\mathcal{L}) \}$$

is recursive, because

$$x \in B \iff \mathbf{Seq}(x) \wedge \ell(x) = 3 \wedge (x)_1 = [\doteq] \wedge (x)_2, (x)_3 \in \ulcorner \text{tm}(\mathcal{L}) \urcorner.$$

Consequently, also $\ulcorner \text{at-Fml}(\mathcal{L}) \urcorner = A \cup B$ is recursive.

$\ulcorner \text{Fml}(\mathcal{L}) \urcorner$ is recursive:

We apply the same strategy as in the proof of the recursiveness of $\ulcorner \text{tm}(\mathcal{L}) \urcorner$. First let us formalize the recursive definition of $\ulcorner \text{Fml}(\mathcal{L}) \urcorner$:

$$x \in \ulcorner \text{Fml}(\mathcal{L}) \urcorner \Leftrightarrow \begin{cases} (x)_2 \in \ulcorner \text{Fml}(\mathcal{L}) \urcorner & \text{if } x = \ulcorner [\neg], (x)_2 \urcorner \\ (x)_2, (x)_3 \in \ulcorner \text{Fml}(\mathcal{L}) \urcorner & \text{if } x = \ulcorner [\rightarrow], (x)_2, (x)_3 \urcorner \\ (x)_2 \in \ulcorner \text{Vbl} \urcorner \wedge (x)_3 \in \ulcorner \text{Fml}(\mathcal{L}) \urcorner & \text{if } x = \ulcorner [\forall], (x)_2, (x)_3 \urcorner \\ x \in \ulcorner \text{at-Fml}(\mathcal{L}) \urcorner & \text{otherwise.} \end{cases}$$

Again we aim at applying 1.4.5 with $F(x) = \mathbb{1}_{\ulcorner \text{Fml}(\mathcal{L}) \urcorner}(x)$ and

$$G(x, v) = \begin{cases} 1 & \text{if } x = \ulcorner [\neg], (x)_2 \urcorner \text{ and } (v)_{(x)_2+1} = 1 \\ 1 & \text{if } x = \ulcorner [\rightarrow], (x)_2, (x)_3 \urcorner \text{ and } (v)_{(x)_2+1} = (v)_{(x)_3+1} = 1 \\ 1 & \text{if } x = \ulcorner [\forall], (x)_2, (x)_3 \urcorner \text{ and } (x)_2 \in \ulcorner \text{Vbl} \urcorner \text{ and } (v)_{(x)_3+1} = 1 \\ 1 & \text{if } x \in \ulcorner \text{at-Fml}(\mathcal{L}) \urcorner \\ 0 & \text{otherwise.} \end{cases}$$

Similar to the proof of (*) above, we see that

$$\mathbb{1}_{\ulcorner \text{Fml}(\mathcal{L}) \urcorner}(x) = G(x, \overline{\mathbb{1}_{\ulcorner \text{Fml}(\mathcal{L}) \urcorner}(x)})$$

and 1.4.5 gives the assertion. \square

4.3. Lemma. *The function $\text{Sub} : \omega^3 \rightarrow \omega$, defined by induction on a via*

$$\text{Sub}(a, b, c) :=$$

$$= \begin{cases} c & \text{if } a \in \ulcorner \text{Vbl} \urcorner \text{ and } a = b \\ \ulcorner (a)_1, \text{Sub}((a)_2, b, c), \dots, \text{Sub}((a)_n, b, c) \urcorner & \text{if } a = \ulcorner (a)_1, \dots, (a)_n \urcorner \text{ with} \\ & n = \ell(a) > 1 \text{ and } (a)_1 \neq [\forall] \\ \ulcorner [\forall], (a)_2, \text{Sub}((a)_3, b, c) \urcorner & \text{if } a = \ulcorner (a)_1, (a)_2, (a)_3 \urcorner, \\ & (a)_1 = [\forall] \text{ and } (a)_2 \neq b \\ a & \text{otherwise} \end{cases}$$

is recursive and codes the substitution of free variables in terms and formulas by other terms. This means, Sub satisfies

$$(\dagger) \quad \text{Sub}(\ulcorner t' \urcorner, \ulcorner x \urcorner, \ulcorner t \urcorner) = \ulcorner t'(x/t) \urcorner \text{ and } \text{Sub}(\ulcorner \varphi \urcorner, \ulcorner x \urcorner, \ulcorner t \urcorner) = \ulcorner \varphi(x/t) \urcorner$$

for all $\varphi \in \text{Fml}(\mathcal{L}), t, t' \in \text{tm}(\mathcal{L})$ and $x \in \text{Vbl}$. Recall that $\varphi(x/t)$ is a formula, even if x is free in φ and not free for t in φ (i.e. if x occurs free at a position in φ that is in the scope of a variable that occurs in t).

Remark: The second case in the definition of Sub covers all steps in the constructions of terms and formulas of the form $F(t_1, \dots, t_n)$, $t_1 \doteq t_2$, $R(t_1, \dots, t_n)$, $\neg\varphi$ and $\varphi \rightarrow \psi$. The case of quantifiers $\forall x \varphi$ is covered in the third and fourth case of the definition of Sub .

Proof. By induction on the complexity of terms and formulas we see directly from the definitions that Sub satisfies the identities in (\dagger) . To see that Sub is recursive

we use 1.4.5 with $F = \text{Sub}$ and $G(a, b, c, v) :=$

$$\begin{cases} c & \text{if } a \in \ulcorner \text{Vbl} \urcorner \text{ and } a = b \\ \prec (a)_1, (v)_{(a)_2+1}, \dots, (v)_{(a)_n+1} \succ & \text{if } a = \prec (a)_1, \dots, (a)_n \succ \text{ with } n = \ell(a) > 1 \\ & \text{and } (a)_1 \neq [\forall] \\ \prec [\forall], (a)_2, (v)_{(a)_3+1} \succ & \text{if } a = \prec (a)_1, (a)_2, (a)_3 \succ, \\ & (a)_1 = [\forall] \text{ and } (a)_2 \neq b \\ a & \text{otherwise} \end{cases}$$

Here, $\prec (a)_1, (v)_{(a)_2+1}, \dots, (v)_{(a)_n+1} \succ$ stands for

$$\mu x \left(\mathbf{Seq}(x) \wedge \ell(x) > 1 \wedge (x)_1 = (a)_1 \wedge \forall 2 \leq i \leq \ell(x) : (x)_i = (v)_{(a)_i+1} \right).$$

Also, note that the condition

$$a = \prec (a)_1, \dots, (a)_n \succ \text{ with } n = \ell(a) > 1 \text{ and } (a)_1 \neq [\forall]$$

is just another way of saying

$$\mathbf{Seq}(a) \wedge \ell(a) > 1 \wedge (a)_1 \neq [\forall].$$

It is straightforward to see that for $a, b, c \in \omega$ with $v = \prec \text{Sub}(0, b, c), \dots, \text{Sub}(a - 1, b, c) \succ$ we have

$$\text{Sub}(a, b, c) = G(a, b, c, v).$$

□

4.4. Lemma. *The following relations of ω are recursive:*

- (i) FR := $\{(\ulcorner \varphi \urcorner, \ulcorner x \urcorner) \mid \varphi \in \text{Fml}(\mathcal{L}) \text{ and } x \in \text{Fr } \varphi\}$.
- (ii) FRSUB := $\{(\ulcorner \varphi \urcorner, \ulcorner x \urcorner, \ulcorner t \urcorner) \mid \varphi \in \text{Fml}(\mathcal{L}), t \in \text{tm}(\mathcal{L}), x \text{ free for } t \text{ in } \varphi\}$.
- (iii) AX := $\{\ulcorner \varphi \urcorner \mid \varphi \in \text{Fml}(\mathcal{L}) \text{ is an axiom of } \mathcal{L}\}$, see definition 2.3.1.
- (iv) MP := $\{(\ulcorner \varphi_1 \urcorner, \ulcorner \varphi_1 \rightarrow \varphi_2 \urcorner, \ulcorner \varphi_2 \urcorner) \mid \varphi_1, \varphi_2 \in \text{Fml}(\mathcal{L})\}$.
- (v) $\ulcorner \text{Sen}(\mathcal{L}) \urcorner$.

Proof. Exercise. This is proved with the same strategy as the claims in 4.2 and 4.3 are proved:

First write out **in detail** the ordinary definition of the "uncoded" objects (e.g. in (i), write out what it means that x is free in φ) using an inductive definition. Then translate this inductive definition into a recursive definition of Gödel numbers. □

5. UNDECIDABILITY AND INCOMPLETENESS

Unless stated otherwise, we continue to work with a recursive language \mathcal{L} as explained at the beginning of section 4.

5.1. **Recursively axiomatizable and decidable theories.**

The results of section 4 put us in a position to formalize various questions on theories and structures from the beginning of our course. For example the question whether a given \mathcal{L} -structure \mathcal{M} has a computable theory: This is grounded as the question on whether the set of all $\ulcorner \varphi \urcorner$ with $\mathcal{M} \models \varphi$ is a recursive subset of ω . Or, we can ask whether a computer program can list all sentences that are true in all groups (which is intuitively true).

5.1.1. Definition. Let $\Sigma \subseteq \text{Fml}(\mathcal{L})$. We call Σ **recursive**, if $\ulcorner \Sigma \urcorner$ is a recursive subset of ω . We call Σ **recursively enumerable**, if $\ulcorner \Sigma \urcorner$ is a recursively enumerable subset of ω .

It should be noticed that for $\varphi \in \text{Fml}(\mathcal{L})$ we have

$$\varphi \in \Sigma \iff \ulcorner \varphi \urcorner \in \ulcorner \Sigma \urcorner,$$

since the coding process is injective on the set of formulas; also see the remark after 4.1.

5.1.2. Definition. Let $\Sigma \subseteq \text{Fml}(\mathcal{L})$. We define

$$\text{Proof}_\Sigma := \{ \langle \ulcorner \varphi_1 \urcorner, \dots, \ulcorner \varphi_n \urcorner \rangle \mid n \in \mathbb{N} \text{ and } (\varphi_1, \dots, \varphi_n) \text{ is a proof from } \Sigma \}.$$

(cf. 2.3.2). The sequence number $\langle \ulcorner \varphi_1 \urcorner, \dots, \ulcorner \varphi_n \urcorner \rangle$ is called the **proof number** of the proof $(\varphi_1, \dots, \varphi_n)$.

Hence for $a \in \omega$ we have

$$\text{Proof}_\Sigma(a) \iff a \text{ is the proof number of a proof from } \Sigma.$$

5.1.3. Proposition. *If $\Sigma \subseteq \text{Fml}(\mathcal{L})$ is recursive, then also Proof_Σ is recursive.*

Proof. For every $a \in \omega$ we have

$$\text{Proof}_\Sigma(a) \iff \mathbf{Seq}(a) \wedge \ell(a) \neq 0 \wedge \forall 1 \leq k \leq \ell(a) \left(\begin{aligned} & ((a)_k \in \ulcorner \Sigma \urcorner \cup \text{AX}) \vee \exists 1 \leq i, j < k : \text{MP}((a)_i, (a)_j, (a)_k) \end{aligned} \right)$$

By 4.4 we get the assertion □

5.1.4. **Definition.** The **deductive closure** $\text{Ded}(\Sigma)$ of a set Σ of \mathcal{L} -sentences is defined as

$$\text{Ded}(\Sigma) = \{\varphi \in \text{Sen}(\mathcal{L}) \mid \Sigma \vdash \varphi\}.$$

Σ is called **deductively closed** if $\Sigma = \text{Ded}(\Sigma)$. Further, an \mathcal{L} -**theory** T in this course is defined to be a consistent and deductively closed set of \mathcal{L} -sentences (so $T \vdash \varphi \Rightarrow \varphi \in T$ for all $\varphi \in \text{Sen}(\mathcal{L})$).³ A **complete theory** is a maximally consistent set of \mathcal{L} -sentences. Convince yourself (the argument is actually identical to the one in propositional logic) that every complete theory is deductively closed, hence is also a theory in our sense. By the completeness theorem, complete theories are exactly the theories of \mathcal{L} -structures, i.e., those subsets of \mathcal{L} -sentences of the form $\text{Th}(\mathcal{M}) = \{\varphi \in \text{Sen}(\mathcal{L}) \mid \mathcal{M} \models \varphi\}$, for some \mathcal{L} -structure \mathcal{M} ; so here $\text{Th}(\mathcal{M})$ is called the **theory of** \mathcal{M} .

5.1.5. **Definition.** Let T be an \mathcal{L} -theory. Every subset Σ of T with $T = \text{Ded}(\Sigma)$ is called an **axiom system of** T . T is called **recursively axiomatizable** if it has a recursive axiom system.

A consistent set Σ of \mathcal{L} -sentences is called **decidable** if $\text{Ded}(\Sigma)$ is recursive, otherwise Σ is called **undecidable**. Hence the theory T is decidable, if T is recursive, otherwise T is undecidable.

The next two theorems give main tools to detect decidable (and complete) theories.

5.1.6. **Theorem.** *The following are equivalent for every \mathcal{L} -theory T :*

- (i) T is recursively axiomatizable.
- (ii) T is recursively enumerable.
- (iii) T has a recursively enumerable axiom system.

Proof.

(i) \Rightarrow (ii). Take $\Sigma \subseteq T$ recursive such that $T = \text{Ded}(\Sigma)$. Then for $a \in \omega$ we have

$$a \in \ulcorner \text{Ded}(\Sigma) \urcorner \iff \exists x \left(\text{Proof}_\Sigma(x) \wedge a = (x)_{\ell(x)} \wedge a \in \ulcorner \text{Sen}(\mathcal{L}) \urcorner \right).$$

By 5.1.3, $\text{Proof}_\Sigma(x)$ is recursive and by 4.4, $\ulcorner \text{Sen}(\mathcal{L}) \urcorner$ is recursive. Since also $(x)_{\ell(x)}$ is recursive, we see that $\ulcorner T \urcorner = \ulcorner \text{Ded}(\Sigma) \urcorner$ is the projection of a recursive set and so it is recursively enumerable.

(ii) \Rightarrow (iii) is trivial.

(iii) \Rightarrow (i). Let $\Sigma \subseteq T$ be recursively enumerable such that $T = \text{Ded}(\Sigma)$.

By 1.5.3 there is a recursive function $f : \omega \rightarrow \omega$ with image $\ulcorner \Sigma \urcorner$. For $n \in \omega$ let $\sigma_n \in \Sigma$ be the sentence with

$$f(n) = \ulcorner \sigma_n \urcorner.$$

Let

$$\gamma_n = \sigma_0 \wedge \sigma_1 \wedge \dots \wedge \sigma_n$$

We define a new function $F : \omega \rightarrow \omega$ by

$$F(n) = \ulcorner \gamma_n \urcorner.$$

³Notice that the definition of what is a theory varies in the logic literature. We take the definition that is most suitable for our purposes.

and check that F is recursive: obviously, there is a recursive function $c : \omega^2 \rightarrow \omega$ such that for all $\varphi, \psi \in \text{Fml}(\mathcal{L})$ we have

$$c(\ulcorner \varphi \urcorner, \ulcorner \psi \urcorner) = \ulcorner \varphi \wedge \psi \urcorner.$$

Since $F(0) = f(0)$ and

$$F(n+1) = c(f(n+1), F(n)),$$

we see that F is recursive.

Define a function $h : \omega \rightarrow \omega$ by

$$h(n) = F(\mu x (\forall i < n : h(i) < F(x))).$$

Then h is recursive (from recursion on previous values) and strictly increasing whose image is contained in $F(\omega)$. Hence by question 13, the image X of h is an infinite recursive subset of $F(\omega)$.

We define

$$\Gamma := \{\gamma_n \mid n \in X\}.$$

Then $\ulcorner \Gamma \urcorner = X$ and so Γ is recursive. Since X is infinite and contained in $F(\omega)$, the choice of the γ_n obviously implies $T = \text{Ded } \Sigma = \text{Ded } \Gamma$. Hence Γ is a recursive axiom system of T , which proves (i). \square

5.1.7. Theorem. *Every complete and recursively axiomatisable theory is decidable.*

Proof. Let T be our theory and let $\Sigma \subseteq T$ be recursive with $T = \text{Ded}(\Sigma)$. By 5.1.6, $\ulcorner T \urcorner$ is recursively enumerable and by the Negation Theorem 1.5.2 from Recursion theory it suffices to show that also $\omega \setminus \ulcorner T \urcorner$ is recursively enumerable.

For $a \in \omega$ we have

$$(*) \quad a \notin \ulcorner T \urcorner \iff a \notin \ulcorner \text{Sen}(\mathcal{L}) \urcorner \vee \prec [\neg], a \succ \in \ulcorner T \urcorner,$$

because T is complete and for $\varphi \in \text{Sen}(\mathcal{L})$ with $a = \ulcorner \varphi \urcorner$ we have $\ulcorner \neg \varphi \urcorner = \prec [\neg], \ulcorner \varphi \urcorner \succ$.

Since T is recursively enumerable it is straightforward to see that also the set $X = \{a \in \omega \mid \prec [\neg], a \succ \in \ulcorner T \urcorner\}$ is recursively enumerable.

Consequently $\omega \setminus \ulcorner T \urcorner$ is the union of two recursive enumerable sets $\omega \setminus \ulcorner \text{Sen}(\mathcal{L}) \urcorner$ and X . However, the union of two recursively enumerable set is readily seen to be recursively enumerable itself. \square

5.1.8. Examples. To see how 5.1.6 and 5.1.7 work together we do two examples. This is not needed later on and we refer to some easy facts from model theory.

- (1) If \mathcal{L} is the language $\{<\}$ of orders and T is the theory of dense linear orders without endpoints, then T is decidable. T is defined as the deductive closure of the set Σ consisting of the following \mathcal{L} -sentences:

- $\forall x \neg x < x$
- $\forall xyz (x < y < z \rightarrow x < z)$
- $\forall xy (x < y \vee x = y \vee y < x)$
- $\forall x \exists yz y < x < z$
- $\forall xy x < y \rightarrow \exists z x < z < y$.

So T is recursively axiomatised, since Σ is finite. Using basic model theory it is not difficult to show that T is indeed complete. So by 5.1.7, T is decidable.

- (2) The theory T of infinite sets in the empty language is decidable. This theory has a recursively enumerable axiom system: Take all the sentences

$$\exists x_1 \dots x_n \bigwedge_{1 \leq i \neq j \leq n} x_i \neq x_j.$$

(This set is indeed recursive, but by 5.1.6 we only need to check that it is recursively enumerable).

Again, basic model theory shows that this theory is complete. So by 5.1.7, T is decidable.

5.2. The first incompleteness theorem.

In this subsection we continue to work with a countable recursive language \mathcal{L} , but now we are assuming that \mathcal{L} extends $\mathcal{L}_{\overline{\omega}}$.

5.2.1. **Lemma.** *The function*

$$\text{Num} : \omega \longrightarrow \omega \text{ defined by } \text{Num}(a) = \ulcorner \underline{a} \urcorner$$

(recall that $\underline{a} = S^a 0$) is recursive.

Proof. We have $\text{Num}(0) = \ulcorner 0 \urcorner$ and $\text{Num}(a+1) = \ulcorner [S], \text{Num}(a) \urcorner$.

Now apply 1.3.3 with $H(b) = \ulcorner [S], b \urcorner$ and initial value $\ulcorner 0 \urcorner$. \square

5.2.2. **Definition.** Let $\Sigma \subseteq \text{Sen}(\mathcal{L})$. We fix a variable w and define $P^\Sigma \subseteq \omega^2$ by

$$P^\Sigma(a, b) \iff \text{Sub}(a, \ulcorner w \urcorner, \text{Num } b) \in \ulcorner \text{Ded } \Sigma \urcorner.$$

Hence for each \mathcal{L} -formula $\varphi(w)$ we have:

$$P^\Sigma(\ulcorner \varphi \urcorner, b) \iff \Sigma \vdash \varphi(w/\underline{b}),$$

because from 4.3 we know $\text{Sub}(\ulcorner \varphi \urcorner, \ulcorner w \urcorner, \text{Num } b) = \ulcorner \varphi(w/\underline{b}) \urcorner$ and $\ulcorner \varphi(w/\underline{b}) \urcorner \in \ulcorner \text{Ded } \Sigma \urcorner$ means $\Sigma \vdash \varphi(\underline{b})$.

So $P^\Sigma(a, b)$ is intended to say that Σ proves the formula that is coded by a when tested at b .

5.2.3. **Proposition.** *If $\Sigma \subseteq \text{Sen}(\mathcal{L})$ such that $\Sigma \cup \Omega$ is consistent, then every recursive subset of ω is of the form*

$$P^\Sigma(a) := \{b \in \omega \mid P^\Sigma(a, b)\}$$

for some $a \in \omega$.

Proof. Let $X \subseteq \omega$ be recursive. From the representability theorem 3.5 we know that X is represented in $\Sigma \cup \Omega$ by some formula $\varphi(w)$. Hence for each $b \in \omega$ we have

$$\begin{aligned} X(b) &\Rightarrow \Sigma \cup \Omega \vdash \varphi(\underline{b}) \text{ and} \\ \neg X(b) &\Rightarrow \Sigma \cup \Omega \vdash \neg \varphi(\underline{b}). \end{aligned}$$

Let $\bigwedge \Omega$ be the conjunction of the nine sentences from Ω . Since $\Sigma \cup \Omega$ is consistent, we get for all $b \in \omega$ that

$$X(b) \iff \Sigma \cup \{\bigwedge \Omega\} \vdash \varphi(\underline{b}).$$

Now this is the same as saying that for all $b \in \omega$

$$X(b) \iff \Sigma \vdash \bigwedge \Omega \rightarrow \varphi(\underline{b}).$$

So if we take $a = \ulcorner \bigwedge \Omega \rightarrow \varphi \urcorner$, then we have for each $b \in \omega$:

$$X(b) \iff \Sigma \vdash \bigwedge \Omega \rightarrow \varphi(\underline{b}) \iff P^\Sigma(a, b),$$

i.e. $X = P^\Sigma(a)$. \square

We will now apply the Cantor anti-diagonal argument to P^Σ . Let us first isolate

5.2.4. Cantor's Anti-diagonal lemma

Let M be any set and let $P \subseteq M \times M$. For $a \in M$ let $P(a) \subseteq M$ be defined by

$$P(a) := \{b \in M \mid P(a, b)\}.$$

Thus, $P(a)$ can be thought of the “the fibre of P above a ” and we have

$$P(a)(b) \iff P(a, b).$$

Let $Q \subseteq M$ be the **anti-diagonal** of P , i.e.

$$Q(b) \iff \neg P(b, b) \quad (b \in M).$$

Then Q is not of the form $P(a)$ for any $a \in M$.

Proof. Suppose Q is $P(a)$ for some $a \in M$. Then for each $b \in M$ we have

$$P(a, b) \iff P(a)(b) \iff Q(b) \iff \neg P(b, b).$$

But this is not possible for $b = a$. □

Remark: This is essentially the argument of the proof that no function $f : M \rightarrow \mathfrak{P}(M)$ (the powerset of M) is surjective: define $P(a, b) \iff b \in f(a)$ for $a, b \in M$; then $P(a)$ is the set $f(a)$, and so by 5.2.4, the subset Q of M of all $b \in M$ with $b \notin f(b)$ is not of the form $f(a)$ for any $a \in M$.

5.2.5. Church's Theorem (Alonso Church, 1936)

If T is an \mathcal{L} -theory such that $T \cup \Omega$ is consistent, then T is undecidable.

Proof. Suppose T is decidable. Then $\ulcorner T \urcorner = \ulcorner \text{Ded } T \urcorner$ is recursive and by definition of P^T , also P^T is recursive.

Let $Q \subseteq \omega$ be the anti-diagonal of P^T . Q is recursive since for $b \in \omega$ we have

$$Q(b) \iff \neg P^T(b, b).$$

By 5.2.3, there is some $a \in \omega$ with $Q = P^T(a)$. But this contradicts Cantor's anti-diagonal lemma 5.2.4. □

If $b = \ulcorner \varphi(w) \urcorner$ for some \mathcal{L} -formula $\varphi(w)$, then the predicate $Q(b)$ in the proof above, says that $\varphi(\underline{b})$ cannot be proved in T .

5.2.6. Gödel's First Incompleteness Theorem (1931, see [Goedel31])

No recursively axiomatizable \mathcal{L} -theory containing Ω is complete.

Proof. Otherwise, by 5.1.7 the theory would also be decidable, in contradiction to 5.2.5. □

Let us have a look at a prominent example.

5.2.7. Definition. (First order Peano arithmetic)

Let **PA** be the set Ω together with all the **induction statements** of \mathcal{L}_ω -formulas in one variable, i.e. all sentences of the form

$$\forall x \left(\left(\varphi(x, 0) \wedge \forall y (\varphi(x, y) \rightarrow \varphi(x, Sy)) \right) \rightarrow \forall y \varphi(x, y) \right),$$

where $\varphi(x, y)$ is an \mathcal{L}_ω -formula with $\text{Fr } \varphi \subseteq \{x_1, \dots, x_n, y\}$, $x := (x_1, \dots, x_n)$.

PA is called **Peano Arithmetic** and is the widely accepted axiom system of number theory.

Obviously $\bar{\omega}$ is a model of **PA**. The axiom system **PA** is incredibly strong (in particular compared to Ω), and before Gödel there was hope that all number theoretic statements were derivable from **PA**.

The set **PA** is recursive, because an integer $n \in \omega$ is a code of an induction statement if and only if there is some $k \leq n$ such that k is the code of a formula $\varphi(x, y)$ as in 5.2.7 and n is the code of the induction statement built from φ .

5.2.8. Corollary. *The set of $\mathcal{L}_{\bar{\omega}}$ -sentences, provable in **PA** is not complete.*

Proof. Since **PA** is recursive, Gödel's First Incompleteness Theorem 5.2.6 applies. \square

Here is another direct consequence of Church's theorem

5.2.9. Corollary. *No $\mathcal{L}_{\bar{\omega}}$ -theory that is satisfied by $\bar{\omega}$ is decidable. In particular $\text{Ded}(\emptyset)$, the set of all absolutely true $\mathcal{L}_{\bar{\omega}}$ -sentences is undecidable.*

Proof. By 5.2.5, as all these theories are consistent with Ω . \square

We now also have a first example of a recursively enumerable set that is not recursive. Corollary 5.2.9 says that the subset $\ulcorner \text{Ded}(\emptyset) \urcorner$ of ω has this property. A more explicit version of such a set will be constructed in 7.3.3.

The Church-Turing thesis revisited

After the statements above one might be tempted to go back to the very beginning and question whether our incarnation of "computable" as *recursive* should be reconsidered. However, a strong argument supporting the Church-Turing thesis is the converse of the representability theorem 3.5:

5.2.10. Converse of the representability theorem

Let T be a recursively axiomatised \mathcal{L} -theory containing Robinson Arithmetic. Then every function and every relation represented in T is recursive.

Proof. Let Σ be a recursive axiom system of T . By 3.2 it suffices to do the case of functions. So let $F : \omega^n \rightarrow \omega$ and let $\varphi(x_1, \dots, x_n, y)$ be an \mathcal{L} -formula such that for all $a_1, \dots, a_n \in \omega$ we have:

$$\Sigma \vdash \forall y \left(\varphi(\underline{a_1}, \dots, \underline{a_n}, y) \leftrightarrow y \doteq \underline{F(a_1, \dots, a_n)} \right).$$

Then

$$(*) \quad F(a_1, \dots, a_n) \text{ is the unique } b \in \omega \text{ with } \Sigma \vdash \varphi(\underline{a_1}, \dots, \underline{a_n}, \underline{b}).$$

We first need to code replacement of variables by several constant terms in a recursive way. This is just an iteration of the function Sub from 4.3. That is: there is a recursive function $\text{Sub}_k : \omega \times \omega^k \times \omega^k \rightarrow \omega$ such that for all $\psi(z, x_1, \dots, x_k, y_1, \dots, y_k) \in \text{Fml}(\mathcal{L})$ and all constant terms t_1, \dots, t_k we have

$$\text{Sub}_k(\ulcorner \psi \urcorner, \ulcorner x_1 \urcorner, \dots, \ulcorner x_k \urcorner, \ulcorner t_1 \urcorner, \dots, \ulcorner t_k \urcorner) = \ulcorner \varphi(x_1/t_1, \dots, x_k/t_k) \urcorner.$$

In particular, the function $\omega^{n+1} \rightarrow \omega$ that maps (a_1, \dots, a_n, b) to

$$\ulcorner \varphi(\underline{a_1}, \dots, \underline{a_n}, \underline{b}) \urcorner$$

is recursive.

We define a function $H : \omega^n \rightarrow \omega$ by

$$H(a_1, \dots, a_n) = \mu z \left(\mathbf{Seq}(z) \wedge \ell(z) = 2 \wedge \text{Proof}_\Sigma((z)_2) \wedge \right. \\ \left. ((z)_2)_{\ell((z)_2)} = \ulcorner \varphi(\underline{a}_1, \dots, \underline{a}_n, \underline{(z)_1} \urcorner \right).$$

So $H(a_1, \dots, a_n)$ is the smallest sequence number of a sequence (z_1, z_2) such that z_2 is the proof number of a proof from Σ that ends with $\varphi(\underline{a}_1, \dots, \underline{a}_n, \underline{z_1})$; observe that by (*) there is actually such a sequence number. Further, by (*), the number z_1 then has to be $F(a_1, \dots, a_n)$ and so

$$F(a_1, \dots, a_n) = (H(a_1, \dots, a_n))_1.$$

The function H is recursive, since $(a_1, \dots, a_n, b) \mapsto \ulcorner \varphi(\underline{a}_1, \dots, \underline{a}_n, \underline{b}) \urcorner$ is recursive, and so F is recursive, too. \square

5.3. Undecidable sentences.

5.3.1. Definition. Let \mathcal{L} be an arbitrary language and let \mathcal{M} be an \mathcal{L} -structure. A subset X of $|\mathcal{M}|^n$ is called **definable in \mathcal{M}** , if there is an \mathcal{L} -formula $\varphi(x_1, \dots, x_n)$ such that

$$X = \{(a_1, \dots, a_n) \in |\mathcal{M}|^n \mid \mathcal{M} \models \varphi[a_1, \dots, a_n]\}.$$

We also write $\varphi[|\mathcal{M}|^n]$ for the set on the right hand side and $\varphi[\mathcal{M}]$, if n is clear from the context.

A function $F : X \rightarrow |\mathcal{M}|^k$ is called definable in \mathcal{M} , if the graph of F (a subset of $|\mathcal{M}|^n \times |\mathcal{M}|^k$) is definable in \mathcal{M} ; observe that in this case also X is definable in \mathcal{M} , because X is the projection of the graph of F to $|\mathcal{M}|^n$.

The subsets of ω^n that are definable in $\bar{\omega}$ (so here the language is $\mathcal{L}_{\bar{\omega}}$) are sometimes called **arithmetic sets**.

5.3.2. Proposition.

- (i) If $X \subseteq \omega^n$ is represented by the formula φ in Ω , then X is defined by φ in $\bar{\omega}$.
- (ii) Every recursively enumerable subset X of ω^n is definable in $\bar{\omega}$.

Proof. (i) By definition we have for all $a_1, \dots, a_n \in \omega$:

$$\begin{aligned} X(a_1, \dots, a_n) &\Rightarrow \Omega \vdash \varphi(\underline{a_1}, \dots, \underline{a_n}) \text{ and} \\ \neg X(a_1, \dots, a_n) &\Rightarrow \Omega \vdash \neg\varphi(\underline{a_1}, \dots, \underline{a_n}). \end{aligned}$$

Since $\bar{\omega} \models \Omega$ we obtain

$$\begin{aligned} X(a_1, \dots, a_n) &\Rightarrow \bar{\omega} \models \varphi(\underline{a_1}, \dots, \underline{a_n}) \text{ and} \\ \neg X(a_1, \dots, a_n) &\Rightarrow \bar{\omega} \models \neg\varphi(\underline{a_1}, \dots, \underline{a_n}). \end{aligned}$$

Thus X is defined in $\bar{\omega}$ by φ .

(ii) By definition, every recursively enumerable subset X of ω^n is the projection of a recursive subset X' of ω^{n+1} . Since recursive sets are representable in Ω , we know from (i) that X' is definable in $\bar{\omega}$ by some formula $\varphi(x_1, \dots, x_n, y)$. Then X is definable in $\bar{\omega}$ by

$$\exists y \varphi.$$

□

In 7.2.3 we shall prove a significant strengthening of 5.3.2(ii).

We return to our general assumption and work with a recursive language \mathcal{L} extending $\mathcal{L}_{\bar{\omega}}$.

5.3.3. Fixed Point Lemma

Let $\varphi(w)$ be an \mathcal{L} -formula. Then there is some \mathcal{L} -sentence ψ such that

$$\Omega \vdash \psi \leftrightarrow \varphi(w / \ulcorner \psi \urcorner).$$

So, the truth of ψ in a model of Ω can be tested in that model by evaluating $\varphi(w)$ at the (numeral of the) Gödel number of ψ .

Proof. Let $F : \omega \rightarrow \omega$ be defined by

$$F(x) = \text{Sub}(x, \ulcorner w \urcorner, \text{Num}(x)).$$

Recall that

$$\text{Sub}(\ulcorner \delta \urcorner, \ulcorner w \urcorner, \text{Num}(a)) = \ulcorner \delta(\underline{a}) \urcorner$$

for all $\delta(w) \in \text{Fml}(\mathcal{L})$ and all $a \in \omega$. Hence for $a = \ulcorner \delta \urcorner$ we get

$$(*) \quad F(\ulcorner \delta \urcorner) = \text{Sub}(\ulcorner \delta \urcorner, \ulcorner w \urcorner, \text{Num}(\ulcorner \delta \urcorner)) = \ulcorner \delta(\ulcorner \delta \urcorner) \urcorner$$

The intuition now is to think of F as a function symbol in the language \mathcal{L} . Then $\varphi(F(w))$ is an \mathcal{L} -formula, call it $\delta(w)$. Then with

$$k = \ulcorner \delta \urcorner = \ulcorner \varphi(F(w)) \urcorner \text{ and } \psi = \varphi(F(k)) = \delta(\ulcorner \delta \urcorner),$$

condition $(*)$ says

$$F(k) = F(\ulcorner \delta \urcorner) = \ulcorner \delta(\ulcorner \delta \urcorner) \urcorner = \ulcorner \varphi(F(k)) \urcorner = \ulcorner \psi \urcorner.$$

Now observe that

$$\psi = \varphi(F(k)) = \varphi(\ulcorner \psi \urcorner) \text{ (ignoring that we don't know } F(k) = \underline{F(k)} \text{)}$$

which gives the assertion.

Let us work this out in detail. The function $F : \omega \rightarrow \omega$ is indeed close to being a function symbol (at least for our purposes), because F is recursive: Hence F is represented in Ω by a formula $\gamma(w, y)$, i.e. for all $a \in \omega$ we have

$$(\dagger) \quad \Omega \vdash \gamma(\underline{a}, y) \leftrightarrow y \doteq \underline{F(a)}.$$

Now the idea of considering $\varphi(F(w))$ is rigorously implemented by taking the formula $\varphi_F(w)$ defined as

$$\exists y (\gamma(w, y) \wedge \varphi(y))$$

Following the idea we define

$$k = \ulcorner \varphi_F(w) \urcorner \text{ and } \psi = \varphi_F(k).$$

Again, $(*)$ says $F(k) = \ulcorner \psi \urcorner$. By (\dagger) for $a = k$ we know that

$$\Omega \vdash \gamma(\underline{k}, y) \leftrightarrow y \doteq \underline{F(k)}.$$

Consequently

$$\Omega \vdash \underbrace{\exists y (\gamma(\underline{k}, y) \wedge \varphi(y))}_{=\psi} \leftrightarrow \exists y (y \doteq \underline{F(k)} \wedge \varphi(y)).$$

On the other hand we obviously have

$$\Omega \vdash \exists y (y \doteq \underline{F(k)} \wedge \varphi(y)) \leftrightarrow \overbrace{\varphi(\underline{F(k)})}^{=\varphi(\ulcorner \psi \urcorner)},$$

proving the assertion. \square

The name Fixed Point Lemma comes from the following consideration. Given an \mathcal{L} -formula $\varphi(w)$ we have a map

$$\Phi : \text{Sen}(\mathcal{L}) \rightarrow \text{Sen}(\mathcal{L}); \psi \mapsto \varphi(\ulcorner \psi \urcorner).$$

If we identify sentences that are provably equivalent w.r.t. Ω , then the Fixed Point Lemma says that this map has a fixed point (although, note that Φ does not induce a map from the equivalence classes to the equivalence classes).

Also observe from the proof of the Fixed Point Lemma that ψ can be explicitly constructed from φ .

The Fixed Point Lemma 5.3.3 is also called **Lemma of self-reference** in the literature.

We have two beautiful applications of the Fixed Point Lemma right away. More will be seen later.

5.3.4. Tarski's Undefinability of Truth

The set of Gödel numbers of $\mathcal{L}_{\bar{\omega}}$ -sentences that are true in $\bar{\omega}$ is not definable in $\bar{\omega}$.

Proof. Let $X = \{\ulcorner \gamma \urcorner \mid \bar{\omega} \models \gamma\}$ be the set in question and suppose X were definable in $\bar{\omega}$. Then also $\omega \setminus X$ is definable in $\bar{\omega}$ by some formula φ (in one free variable).

By the Fixed Point Lemma, there is a sentence ψ with

$$(*) \quad \Omega \vdash \psi \leftrightarrow \varphi(\ulcorner \psi \urcorner).$$

Now take $n = \ulcorner \psi \urcorner$. Since $\bar{\omega} \models \Omega$ we know from (*) that $\bar{\omega} \models \psi \iff \bar{\omega} \models \varphi[n]$.

By choice of φ , $\bar{\omega} \models \varphi[n]$ is equivalent to $\bar{\omega} \models \neg\psi$. So

$$\bar{\omega} \models \psi \iff \bar{\omega} \models \varphi[n] \iff \bar{\omega} \models \neg\psi,$$

a contradiction. \square

For a given recursive (and consistent) subset of $\text{Th}(\bar{\omega})$ we shall now explicitly construct a sentence that is independent of Σ (i.e. neither the sentence nor its negation is provable in Σ). For $\Sigma \subseteq \text{Fml}(\mathcal{L})$ let

$$\text{IsProofOf}_{\Sigma}$$

be the set of all $(a, b) \in \omega^2$ such that $\text{Proof}_{\Sigma \cup \Omega}(a)$ and $(a)_{\ell(a)} = b$. We also write $a \text{ IsProofOf}_{\Sigma} b$ instead of $\text{IsProofOf}_{\Sigma}(a, b)$. Then

$a \text{ IsProofOf}_{\Sigma} b \iff b$ is the Gödel number of a formula and

a is the proof number of a proof of that formula from $\Sigma \cup \Omega$

If Σ is recursive, then $\text{IsProofOf}_{\Sigma}$ is also recursive and so $\text{IsProofOf}_{\Sigma}$ is represented by an $\mathcal{L}_{\bar{\omega}}$ -formula in Ω . We write $\underline{\text{IsProofOf}}_{\Sigma}(x, y)$ for such a formula.

Obviously $\text{IsProofOf}_{\Sigma} = \text{IsProofOf}_{\Sigma \cup \Omega}$ and so $\underline{\text{IsProofOf}}_{\Sigma} = \underline{\text{IsProofOf}}_{\Sigma \cup \Omega}$.

5.3.5. Choice of β_{Σ}

For recursive $\Sigma \subseteq \text{Fml}(\mathcal{L})$ we choose β_{Σ} to be a sentence of \mathcal{L} that satisfies

$$\Omega \vdash \beta_{\Sigma} \leftrightarrow \forall x \neg (\text{IsProofOf}_{\Sigma}(x, \ulcorner \beta_{\Sigma} \urcorner)),$$

according to the Fixed Point Lemma 5.3.3. We will later say more about the possible shape of β_{Σ} , for now we work with an arbitrary but fixed sentence β_{Σ} with the property above. Note that by the proof of the Fixed Point Lemma 5.3.3, it is possible to explicitly construct a concrete sentence that serves as β_{Σ} .

So if $\Sigma \vdash \Omega$, then it is provable in Ω that β_{Σ} asserts its own unprovability in Σ .

An explicit version of Gödel's first incompleteness theorem 5.2.6 for subsets of $\text{Th}(\bar{\omega})$ is the following

5.3.6. Theorem. (Explicit Incompleteness Theorem, Gödel 1931)

If $\Sigma \subseteq \text{Sen}(\mathcal{L})$ is recursive and has a model that expands $\bar{\omega}$ (i.e. a model with universe ω that interprets the $\mathcal{L}_{\bar{\omega}}$ -symbols in the standard way), then β_{Σ} is **independent of Σ** (we also say **undecidable in Σ**), i.e. $\Sigma \not\vdash \beta_{\Sigma}$ and $\Sigma \not\vdash \neg\beta_{\Sigma}$.

(In 7.2.4 we will say more about the shape of β_{Σ}).

Proof. We may of course replace Σ by $\Sigma \cup \Omega$, hence we may assume that $\Omega \subseteq \Sigma$. Then

$$\begin{aligned} \Sigma \vdash \beta_\Sigma &\Rightarrow \Sigma \vdash \forall x \neg (\text{IsProofOf}_\Sigma(x, \ulcorner \beta_\Sigma \urcorner)), \text{ by definition of } \beta_\Sigma \text{ and as } \Omega \subseteq \Sigma \\ &\Rightarrow \Sigma \vdash \neg (\text{IsProofOf}_\Sigma(a, \ulcorner \beta_\Sigma \urcorner)) \text{ for all } a \in \omega \\ &\Rightarrow \neg (a \text{ IsProofOf}_\Sigma \ulcorner \beta_\Sigma \urcorner) \text{ for all } a \in \omega, \text{ because } \text{IsProofOf}_\Sigma \\ &\quad \text{represents } \text{IsProofOf}_\Sigma \text{ in } \Omega \subseteq \Sigma \text{ and } \Sigma \text{ is consistent} \\ &\Rightarrow \Sigma \not\vdash \beta_\Sigma \text{ by definition of } \text{IsProofOf}_\Sigma. \end{aligned}$$

However we also have

$$\begin{aligned} \Sigma \vdash \neg \beta_\Sigma &\Rightarrow \Sigma \vdash \exists x \text{IsProofOf}_\Sigma(x, \ulcorner \beta_\Sigma \urcorner), \text{ by definition of } \beta_\Sigma \text{ and as } \Omega \subseteq \Sigma \\ &\Rightarrow \mathcal{M} \models \text{IsProofOf}_\Sigma(a, \ulcorner \beta_\Sigma \urcorner) \text{ for some } a \in \omega, \text{ where } \mathcal{M} \text{ is a model} \\ &\quad \text{of } \Sigma \text{ that expands } \bar{\omega} \\ &\quad \text{(such an } \mathcal{M} \text{ exists by assumption)} \\ &\Rightarrow \bar{\omega} \models \text{IsProofOf}_\Sigma(a, \ulcorner \beta_\Sigma \urcorner), \text{ because } \mathcal{M} \text{ expands } \bar{\omega} \text{ and} \\ &\quad \text{IsProofOf}_\Sigma \text{ is an } \mathcal{L}_{\bar{\omega}}\text{-formula} \\ &\Rightarrow a \text{ IsProofOf}_\Sigma \ulcorner \beta_\Sigma \urcorner, \text{ because } \text{IsProofOf}_\Sigma \text{ represents} \\ &\quad \text{IsProofOf}_\Sigma \text{ in } \Omega \subseteq \Sigma \\ &\Rightarrow \Sigma \vdash \beta_\Sigma \text{ by definition of } \text{IsProofOf}_\Sigma. \end{aligned}$$

So in either case, we get a contradiction, i.e. β_Σ is undecidable in Σ □

5.3.7. Corollary. *The sentence $\beta_{\mathbf{PA}}$ is independent of Peano Arithmetic.*

Proof. By 5.3.6, since \mathbf{PA} is easily seen to be recursive. □

6. APPLICATIONS TO DECISION PROBLEMS

So far, our results are mainly about the structure $\bar{\omega}$ in the language $\mathcal{L}_{\bar{\omega}}$ and about $\mathcal{L}_{\bar{\omega}}$ -theories satisfied by $\bar{\omega}$. In this section we want to see how we can generalize our results to other structures of other (recursive) languages. An obvious question in this direction is the following: In the structure $\bar{\omega}$, the successor function and the order are already *definable* in the structure $(\omega, +, 1, 0)$ by

$$S(x) = x + 1 \text{ and } x < y \iff \exists z : z \neq 0 \wedge x + z = y.$$

So S and $<$ seem to be redundant and one can ask: Does the structure $(\omega, +, \cdot, 1, 0)$ also have an undecidable theory in the language $\{+, \cdot, 1, 0\}$?

How about the structures $(\mathbb{Z}, +, \cdot)$, $(\mathbb{Q}, +, \cdot)$, $(\mathbb{R}, +, \cdot)$ and $(\mathbb{C}, +, \cdot)$? Do they have an undecidable theory in the language $\{+, \cdot, 1, 0\}$? What about the theory of all fields? Is it decidable?

We can also ask this question about theories and structures, which, on the face of it, have no arithmetic built in, e.g.: Is the theory of partially ordered sets in the language with a single binary relation symbol \leq decidable? Is there a partially order set (X, \leq) which has an undecidable theory?

Finally we can ask: do models of Zermelo-Fraenkel set theory (in some preferred language) have an undecidable theory?

In order to address these questions we need an adequate method to compare structures of different languages with different universes. Such a method is provided by so-called *interpretations*.

6.1. Interpretations.

In this section we will work with arbitrary languages.

6.1.1. Definition. Let \mathcal{L} and \mathcal{L}^+ be languages. An **interpretation of \mathcal{L}^+ in \mathcal{L}** consists of a natural number $d \geq 1$, called the **dimension of the interpretation** as well as the following data:

- I0.** An \mathcal{L} -formula $\mathfrak{U}(v_1, \dots, v_d)$, called the **universe of the interpretation**.
- I1.** An \mathcal{L} -formula $\mathfrak{E}(\bar{x}, \bar{y})$, where \bar{x} and \bar{y} are d -tuples of distinct variables (\mathfrak{E} will be used to interpret the identity of \mathcal{L}^+).
- I2.** For each relation symbol R^+ of \mathcal{L}^+ , an \mathcal{L} -formula $\varphi_{R^+}(\bar{x}_1, \dots, \bar{x}_n)$, where n is the arity of R^+ and $\bar{x}_1, \dots, \bar{x}_n$ are d -tuples of distinct variables.
So if we'd regard the identity symbol of \mathcal{L}^+ as a binary relation symbol, the formula \mathfrak{E} from the previous item could be subsumed here, too.
- I3.** For each function symbol F^+ of \mathcal{L}^+ an \mathcal{L} -formula $\varphi_{F^+}(\bar{x}_1, \dots, \bar{x}_n, \bar{y})$, where n is the arity of F^+ and $\bar{x}_1, \dots, \bar{x}_n, \bar{y}$ are d -tuples of distinct variables.
- I4.** For each constant symbol c^+ of \mathcal{L}^+ an \mathcal{L} -formula $\varphi_{c^+}(\bar{x})$, where \bar{x} is a d -tuple of distinct variables.

If \mathfrak{E} is just equality in \mathcal{L} , (so $\mathfrak{E}(\bar{x}, \bar{y})$ is $\bigwedge_{i=1}^d x_i \doteq y_i$), then we say **definition of \mathcal{L}^+ in \mathcal{L}** instead of interpretation of \mathcal{L}^+ in \mathcal{L} .

If \mathcal{M} is an \mathcal{L} -structure and \mathcal{M}^+ is an \mathcal{L}^+ -structure, then \mathcal{M}^+ is called **interpretable in \mathcal{M}** , if for some interpretation of \mathcal{L}^+ in \mathcal{L} , there is a surjective map

$$\pi : \mathfrak{U}[\mathcal{M}^d] \rightarrow |\mathcal{M}^+|$$

(recall that $\mathfrak{U}[\mathcal{M}^d]$ denotes the set defined by \mathfrak{U}), such that the following hold true:

M1. For $\bar{a}, \bar{b} \in \mathfrak{U}[\mathcal{M}^d]$ we have

$$\mathcal{M} \models \mathfrak{E}[\bar{a}, \bar{b}] \iff \pi(\bar{a}) = \pi(\bar{b}).$$

In particular \mathfrak{E} defines (in \mathcal{M}) an equivalence relation on $\mathfrak{U}[\mathcal{M}^d]$ and the universe of \mathcal{M}^+ can be thought of the set of equivalence classes of \mathfrak{E} .

M2. For each relation symbol R^+ of \mathcal{L}^+ of arity n and all $\bar{a}_1, \dots, \bar{a}_n \in \mathfrak{U}[\mathcal{M}^d]$ we have

$$\mathcal{M} \models \varphi_{R^+}[\bar{a}_1, \dots, \bar{a}_n] \iff \mathcal{M}^+ \models R^+[\pi(\bar{a}_1), \dots, \pi(\bar{a}_n)]$$

M3. For each function symbol F^+ of \mathcal{L}^+ of arity n and all $\bar{a}_1, \dots, \bar{a}_n, \bar{b} \in \mathfrak{U}[\mathcal{M}^d]$ we have

$$\mathcal{M} \models \varphi_{F^+}[\bar{a}_1, \dots, \bar{a}_n, \bar{b}] \iff \mathcal{M}^+ \models \pi(\bar{b}) = F^+(\pi(\bar{a}_1), \dots, \pi(\bar{a}_n)).$$

M4. For each constant symbol c^+ of \mathcal{L}^+ and all $\bar{a} \in \mathfrak{U}[\mathcal{M}^d]$ we have

$$\mathcal{M} \models \varphi_{c^+}[\bar{a}] \iff \mathcal{M}^+ \models \pi(\bar{a}) = c^+.$$

In other words, the equivalence class of \bar{a} w.r.t. $\mathfrak{E}[\mathcal{M}^d]$ is the preimage of $(c^+)^{\mathcal{M}^+}$ under π .

If \mathcal{M}^+ is interpretable in \mathcal{M} and we have chosen an interpretation, then we shall also refer to it as the **interpretation of \mathcal{M}^+ in \mathcal{M}** .

If the interpretation can be chosen such that \mathfrak{E} is just equality, (so $\mathfrak{E}(\bar{x}, \bar{y})$ is $\bigwedge_{i=1}^d x_i \doteq y_i$), then we say \mathcal{M}^+ is **definable in \mathcal{M}** instead of \mathcal{M}^+ is interpretable in \mathcal{M} .

In order to understand this definition one needs to see what is supposed to be achieved with it. We will therefore do some examples. Before we do this we state the goal of the next section, which is supposed to give sufficient reason to analyse interpretability of structures in other structures:

If \mathcal{M} is a structure of a recursive language \mathcal{L} that interprets $\bar{\omega}$ (so here $\mathcal{L}^+ = \mathcal{L}_{\bar{\omega}}$ and $\mathcal{M}^+ = \bar{\omega}$), then every \mathcal{L} -theory that is satisfied by \mathcal{M} , is undecidable.

6.1.2. Examples.

- (1) The structure $(\omega, +, \cdot)$ in the language $\mathcal{L} = \{+, \cdot\}$ defines $\bar{\omega}$ (so here $\mathcal{L}^+ = \mathcal{L}_{\bar{\omega}}$).
- (2) The field \mathbb{R} defines the field \mathbb{C} (so here $\mathcal{L} = \mathcal{L}^+ = \{+, \cdot\}$ and $\mathcal{M}, \mathcal{M}^+$ are the natural structures of the real and the complex field in that language).
- (3) Let \mathcal{L} be the language $\{+, F\}$, where F is a unary function symbol and $+$ is a binary function symbol (as usual). Then the structure $(\omega, f, +)$ where $f(x) := x^2$, defines $\bar{\omega}$ (so here again $\mathcal{L}^+ = \mathcal{L}_{\bar{\omega}}$).
- (4) By Lagrange's 4-squares theorem, every natural number is a sum of 4 squares of natural numbers. Using this we can see that the ring $(\mathbb{Z}, +, \cdot)$ defines $\bar{\omega}$.
- (5) The projective linear group $\text{PGL}(2, \mathbb{R})$ is interpretable in the field \mathbb{R} .
Here we take $\mathcal{L} = \{+, \cdot, 0, 1\}$ and let $\mathcal{L}^+ = \{\cdot, {}^{-1}, 1\}$, where ${}^{-1}$ stands for a unary function symbol. Recall that $\text{PGL}(2, \mathbb{R})$ is the quotient of the group $\text{GL}(2, \mathbb{R})$ of invertible 2×2 -matrices, modulo its center (consisting of all $\lambda \cdot I_2$ with $\lambda \in \mathbb{R} \setminus \{0\}$).
There is nothing special here about \mathbb{R} and the number 2: For every field K , the projective linear group $\text{PGL}(n, K)$ is interpretable in the field K .
- (6) Here are two examples from algebra (for those who are acquainted with this material):
Every commutative domain $(R, +, \cdot)$ interprets its field of fractions.
Generalising (2): every field $(K, +, \cdot)$ defines all its finite extension fields.
- (7) Every model of Zermelo-Fraenkel set theory in the language $\mathcal{L} = \{\varepsilon\}$ (where ε is a binary relation symbol), defines $\bar{\omega}$.

Proof. The details here are left as an exercise. □

6.1.3. **Definition.** If \mathcal{L}^+ is interpreted in \mathcal{L} , then we define a map

$$* : \text{Fml}(\mathcal{L}^+) \longrightarrow \text{Fml}(\mathcal{L})$$

as follows. For every \mathcal{L}^+ -formula φ which has exactly n free variables, φ^* will be an \mathcal{L} -formula in exactly $n \cdot d$ variables, where d is the dimension of the interpretation. We will indicate φ by $\varphi(x_1, \dots, x_n)$ and φ^* by $\varphi^*(\bar{x}_1, \dots, \bar{x}_n)$, i.e. the bar indicates a d -tuple of mutually distinct variables and of course none of the variables in \bar{x}_i should be equal to any of the variables in \bar{x}_j for $i \neq j$. Now for the definition.

$$\begin{aligned} (x \doteq y)^* & \text{ is } \mathfrak{E}(\bar{x}, \bar{y}) \\ (x \doteq c^+)^* \text{ and } (c^+ \doteq x)^* & \text{ are } \varphi_{c^+}(\bar{x}) \\ R^+(x_1, \dots, x_n)^* & \text{ is } \varphi_{R^+}(\bar{x}_1, \dots, \bar{x}_n) \\ (F^+(x_1, \dots, x_n) \doteq y)^* & \text{ is } \varphi_{F^+}(\bar{x}_1, \dots, \bar{x}_n, \bar{y}), \end{aligned}$$

where x, y are variables, c^+ is a constant symbol of \mathcal{L}^+ and R^+, F^+ are relation, function symbols of \mathcal{L} respectively of arity n .

Further, by induction on the complexity of formulas we define \mathcal{L} -formulas $\varphi^*(\bar{x}_1, \dots, \bar{x}_n)$ for every \mathcal{L}^+ -formula $\varphi(x_1, \dots, x_n)$ as follows:

$$\begin{aligned} (\varphi \rightarrow \psi)^* & \text{ is } \varphi^* \rightarrow \psi^* \\ (\varphi \wedge \psi)^* & \text{ is } \varphi^* \wedge \psi^* \\ (\neg\varphi)^* & \text{ is } \neg(\varphi^*) \\ (\exists u\varphi(x_1, \dots, x_n, u))^* & \text{ is } \exists \bar{u} \left(\mathfrak{A}(\bar{u}) \wedge \varphi^*(\bar{x}_1, \dots, \bar{x}_n, \bar{u}) \right) \\ (\forall u\varphi(x_1, \dots, x_n, u))^* & \text{ is } \forall \bar{u} \left(\mathfrak{A}(\bar{u}) \rightarrow \varphi^*(\bar{x}_1, \dots, \bar{x}_n, \bar{u}) \right) \\ (x \doteq t)^* \text{ and } (t \doteq x)^* & \text{ are } \left(\exists y_1 \dots y_n \bigwedge_i y_i \doteq t_i \wedge x = F^+(y_1, \dots, y_n) \right)^* \\ & \text{ where } t = F^+(t_1, \dots, t_n) \\ (s \doteq t)^* & \text{ is } \left(\exists x(x = s \wedge x = t) \right)^* \\ (R^+(t_1, \dots, t_n))^* & \text{ is } \left(\exists y_1 \dots y_n \bigwedge_i y_i \doteq t_i \wedge R^+(y_1, \dots, y_n) \right)^* \end{aligned}$$

6.1.4. **Proposition.** If \mathcal{M}^+ is interpreted in \mathcal{M} and $\varphi(x_1, \dots, x_n)$ is an \mathcal{L}^+ -formula, then for all $\bar{a}_1, \dots, \bar{a}_n \in \mathfrak{A}[\mathcal{M}]$ we have

$$\mathcal{M} \models \varphi^*[\bar{a}_1, \dots, \bar{a}_n] \iff \mathcal{M}^+ \models \varphi[\pi(\bar{a}_1), \dots, \pi(\bar{a}_n)].$$

Proof. This is a straightforward induction on the complexity of φ and is left as an exercise. \square

6.2. Strongly undecidable structures.

6.2.1. Definition. A structure \mathcal{M} of a recursive language \mathcal{L} is called **undecidable**, **decidable** respectively, if its theory $\text{Th}(\mathcal{M})$ has this property. If all theories contained in $\text{Th}(\mathcal{M})$ are undecidable, then \mathcal{M} is called **strongly undecidable**. In other words, a structure is strongly undecidable if it does not satisfy any decidable \mathcal{L} -theory.⁴

By Church's theorem 5.2.5, $\bar{\omega}$ is a strongly undecidable $\mathcal{L}_{\bar{\omega}}$ -structure. In this section we'll show that every structure in a recursive language that interprets $\bar{\omega}$ is also strongly undecidable. First some preparations.

6.2.2. Lemma. *Let \mathcal{L} and \mathcal{L}^+ be recursive languages and let $T \subseteq \text{Sen}(\mathcal{L})$, $T^+ \subseteq \text{Sen}(\mathcal{L}^+)$ be theories.*

Suppose there is a recursive function $F : \omega \rightarrow \omega$ such that for every $\varphi \in \text{Sen}(\mathcal{L}^+)$, there is a sentence $\varphi^ \in \text{Sen}(\mathcal{L})$ with $F(\ulcorner \varphi \urcorner) = \ulcorner \varphi^* \urcorner$ such that*

$$T^+ \vdash \varphi \iff T \vdash \varphi^*.$$

Then

$$T^+ \text{ is undecidable} \Rightarrow T \text{ is undecidable.}$$

Proof. For every $n \in \omega$ we have

$$n \in \ulcorner T^+ \urcorner \iff n \in \ulcorner \text{Sen}(\mathcal{L}^+) \urcorner \wedge F(n) \in \ulcorner T \urcorner.$$

Hence if $\ulcorner T \urcorner$ is recursive, then also $\ulcorner T^+ \urcorner$ is recursive. \square

6.2.3. Lemma and Definition. *Let \mathcal{L} and \mathcal{L}^+ be recursive languages. An interpretation of \mathcal{L}^+ in \mathcal{L} is **recursive** if there is a recursive function $f : \omega \rightarrow \omega$ such that for every non-logical symbol s of \mathcal{L}^+ (hence s is a constant, a relation or a function symbol of \mathcal{L}^+) we have*

$$f(\ulcorner s \urcorner) = \ulcorner \varphi_s \urcorner$$

(see 6.1.1 for the definition of φ_s). Observe that this is always the case if \mathcal{L}^+ is finite.

Now suppose we are given a recursive interpretation of \mathcal{L}^+ in \mathcal{L} and we have chosen \mathcal{L} -formulas φ^ as in 6.1.3 for every \mathcal{L}^+ -formula φ . Then there is a recursive function $F : \omega \rightarrow \omega$ such that for every $\varphi \in \text{Sen}(\mathcal{L}^+)$ we have $F(\ulcorner \varphi \urcorner) = \ulcorner \varphi^* \urcorner$.*

Proof. The existence of F is straightforward: Just write out the recursive definition of the φ^* in 6.1.3, in terms of Gödel numbers and apply recursion on previous values. \square

6.2.4. Proposition. *If T is a decidable theory in a language \mathcal{L} and $\Sigma \subseteq \text{Sen}(\mathcal{L})$ is finite, then also $\text{Ded}(T \cup \Sigma)$ is decidable.*

Proof. Let $T_1 := \text{Ded}(T \cup \Sigma)$ and let $\sigma \in \text{Sen}(\mathcal{L})$ be the conjunction of the sentences in Σ .

For $\varphi \in \text{Sen}(\mathcal{L})$ we have (by the deduction theorem)

$$T \cup \Sigma \vdash \varphi \iff T \vdash \sigma \rightarrow \varphi.$$

⁴An example of an undecidable structure that is not strongly undecidable is the field \mathcal{M} of real numbers together with a named non-recursive real number r in the language $\{+, \cdot, c\}$, where c is interpreted as r . The deductive closure of the $\{+, \cdot\}$ -theory of $(\mathbb{R}, +, \cdot)$ in \mathcal{L} is decidable, but \mathcal{M} is not decidable. Proofs are omitted here.

Hence

$$\ulcorner \varphi \urcorner \in \ulcorner T_1 \urcorner \iff \ulcorner \sigma \rightarrow \varphi \urcorner \in \ulcorner T \urcorner.$$

Consequently, for arbitrary $a \in \omega$ we have

$$a \in \ulcorner T_1 \urcorner \iff a \in \ulcorner \text{Sen}(\mathcal{L}) \urcorner \wedge \left(\ulcorner \prec [\rightarrow], \ulcorner \sigma \urcorner, a \succ \urcorner \in \ulcorner T \urcorner \right).$$

Since $\ulcorner T \urcorner$ is decidable, we see that $\ulcorner T_1 \urcorner$ is decidable as well. \square

6.2.5. Tarski's theorem on strongly undecidable structures

Let \mathcal{L} and \mathcal{L}^+ be recursive languages and suppose we are given a recursive interpretation of \mathcal{L}^+ in \mathcal{L} . Let \mathcal{M}^+ be an \mathcal{L}^+ -structure and let \mathcal{M} be an \mathcal{L} -structure that interprets \mathcal{M}^+ according to this interpretation. Then

- (i) If \mathcal{M}^+ is undecidable, then also \mathcal{M} is undecidable.
- (ii) If \mathcal{M}^+ is strongly undecidable and \mathcal{L}^+ is finite, then also \mathcal{M} is strongly undecidable.

Proof. (i) We know this already from 6.2.2 applied to $\text{Th}(\mathcal{M})$ and $\text{Th}(\mathcal{M}^+)$. The required assumptions are satisfied by 6.2.3 and 6.1.4.

(ii) Let T be an \mathcal{L} -theory with $\mathcal{M} \models T$. We need to show that T is undecidable. The key point is that we can express the interpretation of \mathcal{M}^+ in \mathcal{M} by a finite set INT of \mathcal{L} -sentences that are true in \mathcal{M} . Then, for every model \mathcal{N} of $T \cup \text{INT}$, the truth of INT in \mathcal{N} allows us to interpret an \mathcal{L}^+ -structure \mathcal{N}^+ , just in the same way as \mathcal{M}^+ is interpreted in \mathcal{M} . We then look at the \mathcal{L}^+ -theory T^+ , axiomatised by all \mathcal{L}^+ -sentences for which $T \cup \text{INT} \vdash \varphi^*$ and verify that \mathcal{M}^+ is a model of it. So by assumption, T^+ is undecidable. We will then use 6.2.2 to show that also $\text{Ded}(T \cup \text{INT})$ is undecidable. Since INT is finite, we may finally use 6.2.4 to deduce that also T is undecidable.

Now let us carry this out in detail. INT consist of the following \mathcal{L} -sentences (written in a human readable way):

- (1) $\exists \bar{x} \mathfrak{U}(\bar{x})$.
- (2) The \mathcal{L} -sentence saying that " $\mathfrak{E}(\bar{x}, \bar{y})$ defines an equivalence relation of d -tuples realizing $\mathfrak{U}(\bar{x})$ ".
- (3) For every n -ary relation symbol R^+ of \mathcal{L}^+ , the \mathcal{L} -sentence that expresses " $\varphi_{R^+}(\bar{x}_1, \dots, \bar{x}_n)$ defines on the set of n -tuples of d -tuples from \mathfrak{U} , a set of equivalence classes w.r.t. \mathfrak{E} ."

This means we take the universal closure of

$$\varphi_{R^+}(\bar{x}_1, \dots, \bar{x}_n) \wedge \bigwedge_{i=1}^n \mathfrak{U}(\bar{x}_i) \wedge \mathfrak{U}(\bar{y}_i) \rightarrow \left(\varphi_{R^+}(\bar{y}_1, \dots, \bar{y}_n) \leftrightarrow \bigwedge_{i=1}^n \mathfrak{E}(\bar{x}_i, \bar{y}_i) \right)$$

- (4) For every n -ary function symbol F^+ of \mathcal{L}^+ the \mathcal{L} -sentence that expresses " $\varphi_{F^+}(\bar{x}_1, \dots, \bar{x}_n, \bar{y})$ defines the (preimage of the) graph of a function from n -tuples of equivalence classes defined by \mathfrak{E} on \mathfrak{U} , to these equivalence classes."
- (5) For every constant symbol c^+ of \mathcal{L}^+ the \mathcal{L} -sentence saying that $\varphi_{c^+}(\bar{x})$ is an equivalence class of \mathfrak{E} on \mathfrak{U} .

Observe that INT is finite by assumption on \mathcal{L}^+ . Since \mathcal{M}^+ is interpretable in \mathcal{M} we certainly know $\mathcal{M} \models \text{INT}$.

Now let us take a model \mathcal{N} of INT and define an \mathcal{L} -structure \mathcal{N}^+ as follows:

- The universe of \mathcal{N}^+ is the set of equivalence classes of $\mathfrak{U}[\mathcal{N}^d]$ modulo $\mathfrak{E}[\mathcal{N}^{2d}]$.

- The non-logical symbols of \mathcal{L}^+ are interpreted in \mathcal{N}^+ following the conditions (1)-(5) above, e.g. $(F^+)^{\mathcal{N}^+}$ maps the equivalence class of $(\bar{a}_1, \dots, \bar{a}_n)$ to the equivalence class of any element $\bar{b} \in \mathfrak{U}[\mathcal{N}^d]$ satisfying

$$\mathcal{N} \models \varphi_{F^+}(\bar{a}_1, \dots, \bar{a}_n, \bar{b}).$$

Since $\mathcal{N} \models (4)$, this indeed defines the graph of a function $|\mathcal{N}^+|^n \rightarrow |\mathcal{N}^+|$. By choice of INT, we then know that \mathcal{N} interprets \mathcal{N}^+ , with the same interpretation used to interpret \mathcal{M}^+ in \mathcal{M} . Note that there is no conflict in the notation here because the original structure \mathcal{M}^+ is isomorphic to the structure just defined for \mathcal{M} (the isomorphism maps $\pi(\bar{a})$ to the equivalence class of \bar{a}).

In particular we may apply 6.1.4 and get for every \mathcal{L}^+ -sentence φ and every model \mathcal{N} of INT:

$$(\dagger) \quad \mathcal{N} \models \varphi^* \iff \mathcal{N}^+ \models \varphi.$$

Now let

$$\Sigma^+ = \{\varphi \in \text{Sen}(\mathcal{L}^+) \mid T \cup \text{INT} \vdash \varphi^*\}.$$

Claim. For every model \mathcal{N} of $T \cup \text{INT}$ we have $\mathcal{N}^+ \models \Sigma^+$.

To see this, take $\varphi \in \Sigma^+$. By definition of Σ^+ we know $T \cup \text{INT} \vdash \varphi^*$. Since $\mathcal{N} \models T \cup \text{INT}$ we get $\mathcal{N} \models \varphi^*$ and by the implication \Rightarrow in (\dagger) we see $\mathcal{N}^+ \models \varphi$.

Having proved the claim we now show

$$(+) \quad \Sigma^+ \vdash \varphi \iff T \cup \text{INT} \vdash \varphi^*$$

for all \mathcal{L}^+ -sentences φ .

The implication \Leftarrow is trivial by definition of Σ^+ . The implication \Rightarrow follows from (\dagger) and the claim (using the completeness theorem): Assume $\Sigma^+ \vdash \varphi$ and take a model \mathcal{N} of $T \cup \text{INT}$. Then $\mathcal{N}^+ \models \Sigma^+$ by the claim, and so $\mathcal{N}^+ \models \varphi$. Now from the implication \Leftarrow in (\dagger) we get $\mathcal{N} \models \varphi^*$ as required.

Having proved $(+)$ we may finally apply 6.2.2:

- By the claim, we know that $\mathcal{M}^+ \models \text{Ded}(\Sigma^+)$ and so by assumption, $\text{Ded}(\Sigma^+)$ is undecidable.
- By 6.2.3, there is a recursive function $F : \omega \rightarrow \omega$ such that for every $\varphi \in \text{Sen}(\mathcal{L}^+)$ we have $F(\ulcorner \varphi \urcorner) = \ulcorner \varphi^* \urcorner$ (we knew this all along, note that the existence of this function only depends on the recursive interpretation of \mathcal{L}^+ in \mathcal{L}).
- By $(+)$ we have $\Sigma^+ \vdash \varphi \iff T \cup \text{INT} \vdash \varphi^*$ for all $\varphi \in \text{Sen}(\mathcal{L}^+)$.

So by 6.2.2, $\text{Ded}(T \cup \text{INT})$ is undecidable. Since INT is finite we know from 6.2.4 that T is undecidable, too. \square

6.2.6. Corollary. *The following structures (in their corresponding languages) are strongly undecidable:*

- (i) *The structure $(\omega, +, \cdot)$ in the language $\{+, \cdot\}$.*
- (ii) *The structure $(\mathbb{Z}, +, \cdot)$ in the language $\{+, \cdot\}$. Consequently, the theories of rings, of commutative rings and of integral domains are all undecidable. Also the theory of ordered rings in the language $\{+, \cdot, \leq\}$ is undecidable (it has $(\mathbb{Z}, +, \cdot, \leq)$ as a model).*
- (iii) *The structure $(\omega, f, +)$ in the language $\{F, +\}$, where F is a unary function symbol and $f(x) := x^2$.*

(iv) Every model of ZF in the language with a binary relation symbol. Consequently, ZF is undecidable.

Proof. By 6.1.2, successively we see that each of these structures actually define a strongly undecidable structure. Hence 6.2.5 applies. \square

6.2.7. Corollary. Let \mathcal{L} be a language containing a binary relation symbol. The set of all sentences φ with $\vdash \varphi$ is undecidable.

Proof. Our language possess a strongly undecidable structure: every model of ZF . \square

We'll now record other celebrated undecidability results, they all use serious methods from algebra and geometry to interpret the standard model.

6.2.8. Theorem. (Julia Robinson, see [Robinson])
The field $(\mathbb{Q}, +, \cdot)$ defines the set \mathbb{Z} .

6.2.9. Corollary. The field $(\mathbb{Q}, +, \cdot)$ is strongly undecidable and consequently also the theory of fields (of characteristic 0) in the language $\{+, \cdot\}$ is undecidable.

Proof. By 6.2.6(ii) we know that $(\mathbb{Z}, +, \cdot)$ is strongly undecidable. By J. Robinson's theorem, $(\mathbb{Q}, +, \cdot)$ defines $(\mathbb{Z}, +, \cdot)$. Hence by 6.2.5, also $(\mathbb{Q}, +, \cdot)$ is strongly undecidable. \square

In fact J. Robinson also showed that all number fields (i.e. finite extensions of \mathbb{Q}) define the set \mathbb{Z} and are therefore all undecidable.

Finally a strongly undecidable structure that is a priori remote from the standard model:

6.2.10. Theorem. (A. Grzegorzcyk, see [Grzegorzcyk])

The partially ordered set (A, \leq) of all closed subsets of \mathbb{R}^2 (so the universe is the set of closed subsets of \mathbb{R}^2 and \leq is to be understood as inclusion) interprets the standard model $\bar{\omega}$. In particular the following theories of partially ordered sets are all undecidable: Heyting algebras, (distributive) lattices, partially ordered sets itself.

On the other hand the following fields are all decidable (in the language $\{+, \cdot\}$):

- The field of complex numbers (A. Robinson and in its algebraic geometric form earlier by Chevalley)
- The field of real numbers (A. Tarski)
- The field of p-adic numbers (A. Macintyre)

Further, in contrast to Grzegorzcyk's theorem, Tarski has shown that the theory of boolean algebras is decidable.

7. THE ARITHMETIC HIERARCHY

In this section, \mathcal{L} always denotes the language $\mathcal{L}_\omega = \{<, +, \cdot, S, 0\}$ of arithmetic.

7.1. The structure of arithmetic formulas.

7.1.1. Definition. A Δ_0 -**formula** is an \mathcal{L} -formula whose only quantifiers are bounded by terms of \mathcal{L} . This means: Δ_0 is the smallest set of \mathcal{L} -formulas containing the atomic \mathcal{L} -formulas that is closed under boolean connectives and bounded universal quantification; explicitly:

$$\begin{aligned} \varphi, \psi \in \Delta_0 &\implies \neg\varphi, \varphi \rightarrow \psi \in \Delta_0 \text{ and} \\ \varphi \in \Delta_0, t \in \text{tm}(\mathcal{L}), x \notin \text{Fr}(t) &\implies \forall x < t \varphi \in \Delta_0. \end{aligned}$$

A Σ_1 -**formula** is a formula of the form

$$\exists x_1 \dots \exists x_k \delta,$$

where δ is a Δ_0 -formula and $k \geq 0$.

A Π_1 -**formula** is a formula of the form

$$\forall x_1 \dots \forall x_k \delta,$$

where δ is a Δ_0 -formula and $k \geq 0$.

So by definition $\Delta_0 \subseteq \Sigma_1 \cap \Pi_1$.

More generally we define for $m \geq 1$, sets Σ_m, Π_m of \mathcal{L} -formulas as follows:

Σ_m is the set of all \mathcal{L} -formulas of the form

$$\exists \bar{x}_m \forall \bar{x}_{m-1} \dots \delta$$

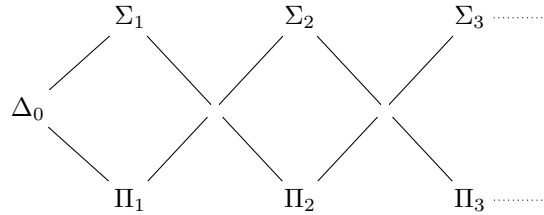
where δ is a Δ_0 -formula $\bar{x}_1, \dots, \bar{x}_m$ are finite (and possibly empty) tuples of variables; here, $\forall \bar{x}$ is shorthand for $\forall x_1 \dots \forall x_k$, if $\bar{x} = (x_1, \dots, x_k)$; also, the clause "and possibly empty" means that the corresponding block of quantifiers possibly does not occur.

Π_m is the set of all \mathcal{L} -formulas of the form

$$\forall \bar{x}_m \exists \bar{x}_{m-1} \dots \delta$$

where δ is a Δ_0 -formula $\bar{x}_1, \dots, \bar{x}_m$ are finite (and possibly empty) tuples of variables.

It follows that $\Sigma_m, \Pi_m \subseteq \Sigma_{m+1}, \Pi_{m+1}$. In a diagram:



For notational issues we also define $\Sigma_0 = \Pi_0 = \Delta_0$.

7.1.2. Definition. If T is a set of \mathcal{L} -formulas, then we write

$$\Delta_0(T) = \{\varphi \in \text{Fml}(\mathcal{L}) \mid \text{there is some } \delta \in \Delta_0 \text{ with } T \vdash \varphi \leftrightarrow \delta\}.$$

Similarly, $\Sigma_m(T)$, $\Pi_m(T)$ denote the set of all \mathcal{L} -formulas for which there exists some $\psi \in \Sigma_m$, Π_m respectively with $T \vdash \varphi \leftrightarrow \psi$.

7.1.3. Remarks.

- (i) $\Delta_0(T)$ is obviously closed under boolean connectives and bounded quantification (i.e. bounded universal and bounded existential quantification).
- (ii) If φ is an \mathcal{L} -formula, then clearly

$$\varphi \in \Sigma_m(T) \iff \neg\varphi \in \Pi_m(T)$$

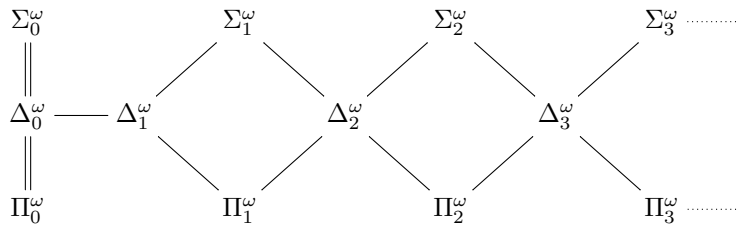
- (iii) $\Sigma_m(T)$ and $\Pi_m(T)$ are closed under finite conjunctions and disjunctions: Just move all the quantifiers in front and possibly change variables to avoid clash of variables.
- (iv) It is a routine exercise to see that all sets Σ_m, Π_m are recursive. However, the sets $\Sigma_m(T)$ and $\Pi_m(T)$ are in general *not* recursive if T is recursive (e.g. if $T = \emptyset$).

7.1.4. Definition. We define

$$\begin{aligned} \Delta_0^\omega &= \{\delta[\bar{\omega}] \subseteq \omega^n \mid n \in \mathbb{N}, \delta(x_1, \dots, x_n) \in \Delta_0\}, \\ \Sigma_m^\omega &= \{\varphi[\bar{\omega}] \subseteq \omega^n \mid n \in \mathbb{N}, \varphi(x_1, \dots, x_n) \in \Sigma_m\} \text{ and} \\ \Pi_m^\omega &= \{\varphi[\bar{\omega}] \subseteq \omega^n \mid n \in \mathbb{N}, \varphi(x_1, \dots, x_n) \in \Pi_m\}. \end{aligned}$$

Recall that $\varphi[\bar{\omega}] = \{a \in \omega^n \mid \bar{\omega} \models \varphi[a_1, \dots, a_n]\}$, where $\varphi(x_1, \dots, x_n)$ is an \mathcal{L} -formula. We say that a set $X \subseteq \omega^n$ is Σ_m (resp. Π_m), if $X \in \Sigma_m^\omega$ (resp. $X \in \Pi_m^\omega$).

The intersection of Σ_m^ω and Π_m^ω is denoted by Δ_m^ω . A subset X of ω^n is called Σ_m , Π_m or Δ_m if X is in Σ_m^ω , in Π_m^ω or in Δ_m^ω . We have the following diagram of inclusions:



7.1.5. Remark. Every set in Δ_0^ω is primitive recursive.

Proof. Clearly every atomic formula defines a primitive recursive set. Hence every set in Δ_0^ω is obtained from primitive recursive sets by boolean combinations and bounded quantification and so all sets in Δ_0^ω are primitive recursive. \square

Our goal in this subsection is to show that the sets $\Sigma_m(T)$, $\Pi_m(T)$ are closed under bounded quantification (cf. 7.1.8). In order to see this we have to swap a bounded universal quantifier with an existential quantifier (modulo T). This cannot be done in $T = \Omega$. What is needed is the following axiom schema.

7.1.6. Definition. The **collection schema** is the following set of \mathcal{L} -sentences. For any \mathcal{L} -formula $\varphi(x, z, \bar{u})$ the sentence

$$\forall \bar{u} \forall y \left((\forall x < y \exists z \varphi(x, z, \bar{u})) \rightarrow \exists w \forall x < y \exists z < w \varphi(x, z, \bar{u}) \right)$$

To explain the collection schema, first note that the variables \bar{u} here act as place holder for parameters. If φ is just $\varphi(x, z)$, then the sentence in the collection schema defined for φ says the following in a model: Given a bound y , if $\varphi(n, z)$ is solvable for all $n < y$, then there is a bound w where we can find solutions. So obviously, the collection schema holds true in \bar{w} . We need to do better and show:

7.1.7. Proposition. *The collection schema is provable in Peano Arithmetic.*

Proof. Take an \mathcal{L} -formula $\varphi(x, z, \bar{u})$. We apply **PA** (cf. 5.2.7) to the formula $\psi(\bar{u}, y)$ defined as

$$(\forall x < y \exists z \varphi(x, z, \bar{u})) \rightarrow \exists w \forall x < y \exists z < w \varphi(x, z, \bar{u}).$$

So we show that in every model \mathcal{M} of **PA**, and every choice of a \bar{u} -tuple \bar{a} from $|\mathcal{M}|$ and all $b \in |\mathcal{M}|$ we have $\mathcal{M} \models \psi[\bar{a}, 0]$ and $\mathcal{M} \models \psi[\bar{a}, b] \rightarrow \psi[\bar{a}, S^{\mathcal{M}}(b)]$. Once this is confirmed, **PA** tells us that $\mathcal{M} \models \forall y \psi[\bar{a}]$. Since \bar{a} was arbitrary, this means $\mathcal{M} \models \forall \bar{u} \forall y \psi$, as required.

The first condition, $\mathcal{M} \models \psi[\bar{a}, 0]$ is trivially true, since **PA** contains Ω and Ω contains $\nexists x x < 0$.

To see the second condition we assume that $\mathcal{M} \models \psi[\bar{a}, b]$ and show that $\mathcal{M} \models \psi[\bar{a}, S^{\mathcal{M}}(b)]$. So assume $\mathcal{M} \models \forall x < S^{\mathcal{M}}(b) \exists z \varphi(x, z, \bar{a})$. Then also $\mathcal{M} \models \forall x < b \exists z \varphi(x, z, \bar{a})$ and from $\mathcal{M} \models \psi[\bar{a}, b]$ we get some $c \in |\mathcal{M}|$ such that

$$\mathcal{M} \models \forall x < b \exists z < c \varphi(x, z, \bar{a}).$$

Further, there is some $c' \in |\mathcal{M}|$ with $\mathcal{M} \models \varphi(b, c', \bar{a})$. Now by $\Omega 9$ we have $c < c', c = c'$ or $c' < c$. We take $d = S^{\mathcal{M}}(c')$ if $c < c'$ and $d = S^{\mathcal{M}}(c)$ otherwise. Then $c' < d$ and for each $e \in |\mathcal{M}|$ with $e < c$ we also know $e < d$ (note that **PA** implies that $<$ is a total order in all models). Therefore

$$\mathcal{M} \models \forall x < S^{\mathcal{M}}(b) \exists z < d \varphi(x, z, \bar{a}),$$

as required. \square

7.1.8. Proposition. *If T is a consistent set of \mathcal{L} -sentences containing **PA**, then $\Sigma_m(T)$ and $\Pi_m(T)$ are closed under bounded quantifications.*

Proof. Since $\varphi \in \Sigma_m(T) \iff \neg \varphi \in \Pi_m(T)$ for all \mathcal{L} -formulas φ , it suffices to show that $\Sigma_m(T)$ is closed under bounded universal quantification (for bounded existential quantification note that $\exists x < t \varphi$ is equivalent to $\exists x (x < t \wedge \varphi) \in \Sigma_m(T)$, cf. 7.1.3).

Take an \mathcal{L} -term t not containing the variable x and let $\varphi \in \Sigma_m(T)$. We must find some $\psi \in \Sigma_m(T)$ with $T \vdash \psi \leftrightarrow \forall x < t \varphi$. We do an induction on m , where $m = 0$ is obvious, because $\Delta_0(T)$ is closed under taking bounded quantification.

We may assume that φ is of the form

$$\exists x_1 \dots x_k \varphi_0 \text{ with } \varphi_0 \in \Pi_{m-1}$$

and do an induction by k . The case $k = 0$ is clear. By 7.1.7, **PA** proves the collection schema. As T contains **PA** we know that

$$T \vdash \forall x < y \exists x_k \exists x_1 \dots x_{k-1} \varphi_0 \leftrightarrow \exists w \forall x < y \exists x_k < w \exists x_1 \dots x_{k-1} \varphi_0.$$

Further we have

$$T \vdash \exists x_k < w \exists x_1 \dots x_{k-1} \varphi_0 \leftrightarrow \exists x_1 \dots x_{k-1} (\exists x_k < w \varphi_0)$$

By induction on m we know that $\exists x_k < w \varphi_0 \in \Pi_{m-1}(T)$. Hence by induction on k we know $\forall x < y \exists x_k < w \exists x_1 \dots x_{k-1} \varphi_0 \in \Sigma_m(T)$, as required. \square

7.2. Recursion revisited.

7.2.1. **Theorem.** *Let φ be an \mathcal{L} -sentence.*

- (i) *If $\varphi \in \Delta_0$ then $\Omega \vdash \varphi$ or $\Omega \vdash \neg \varphi$, i.e. Ω is complete for Δ_0 -sentences.*
- (ii) *If $\varphi \in \Sigma_1$, then $\Omega \vdash \varphi \iff \bar{\omega} \models \varphi$.*

Proof. (i) Recall from 3.3 that for every model \mathcal{M} of Ω , there is a unique embedding $\bar{\omega} \rightarrow \mathcal{M}$. Hence if φ is a quantifier free sentence and $\bar{\omega} \models \varphi$, then $\Omega \vdash \varphi$.

Further, it is clear that the asserted property holds true for $\neg \varphi$ and for $\varphi \wedge \psi$ if it holds for φ and ψ (if $\Omega \vdash \neg \varphi$, then clearly $\Omega \vdash \neg(\varphi \wedge \psi)$). So by induction on the number of quantifiers it remains to assume that φ is of the form $\forall x < t \psi$ for some \mathcal{L} -formula $\psi(x)$ and that we know our assertion for sentences with fewer quantifiers than φ (we cannot assume the assertion for $\psi(x)$, because $\psi(x)$ is in general not a sentence).

Since t is a closed term (observe that φ is assumed to be a sentence), we infer from 3.3 that $\Omega \vdash t \doteq \underline{n}$ for some $n \in \omega$.

Hence we may assume that φ is $\forall x < \underline{n} \psi$. Now by 3.4 we know that

$$\Omega \vdash x < \underline{n} \leftrightarrow (x \doteq 0 \vee \dots \vee x \doteq \underline{n}).$$

Consequently,

$$\Omega \vdash (\forall x < \underline{n} \psi) \leftrightarrow \psi(0) \wedge \dots \wedge \psi(\underline{n-1}).$$

But the sentence on the right hand side has fewer quantifiers than φ and so we get the assertion for φ as required.

(ii) The implication \Rightarrow is clear because $\bar{\omega} \models \Omega$. For the converse, write φ as

$$\exists x_1 \dots x_k \delta \text{ with } \delta \in \Delta_0$$

As $\bar{\omega} \models \varphi$, there are $n_1, \dots, n_k \in \omega$ with $\bar{\omega} \models \delta[n_1, \dots, n_k]$. But this means $\bar{\omega} \models \delta(\underline{n_1}, \dots, \underline{n_k})$. Now $\delta(\underline{n_1}, \dots, \underline{n_k}) \in \Delta_0$ and $\bar{\omega} \models \Omega$. So by (i) we know $\Omega \vdash \delta(\underline{n_1}, \dots, \underline{n_k})$ and thus $\Omega \vdash \varphi$. \square

Next we have a closer look at the representability theorem 3.5 for **PA**.

7.2.2. **Theorem.** *Let T be a set of \mathcal{L} -sentences with $T \vdash \mathbf{PA}$. Then every recursive function and every recursive predicate is represented in T by a Σ_1 -formula and by a Π_1 -formula.*

Proof. We first deal with functions.

From claim 3.5.1 in the proof of 3.5 we know that all elementary functions from **R1** of definition 1.1.1 are represented in Ω by a quantifier free \mathcal{L} -formula. In particular, these functions are represented in T by a Σ_1 -formula and by a Π_1 -formula.

By claim 3.5.2 in the proof of 3.5, the composition of functions that are represented in T by Σ_1 -formulas and by Π_1 -formulas is again represented in T by a Σ_1 -formula and by a Π_1 -formula.

Finally, a function F obtained by μ -recursion from a function G that is represented in T by a Σ_1 -formula and by a Π_1 -formula, is itself represented in T by a Σ_1 -formula and by a Π_1 -formula: Here we use claim 3.5.3 in the proof of 3.5, which says that F is represented in T by a formula ψ that is obtained from a formula representing G using a bounded universal quantifier.

Now as T proves **PA** we know from 7.1.8, that $\Sigma_1(T)$ and $\Pi_1(T)$ are closed under bounded universal quantification. Thus, $\psi \in \Sigma_1(T)$ and $\psi \in \Pi_1(T)$.

This shows the theorem for functions. For relations P we may use the result on functions and notice that P is represented in T by $\varphi(\bar{x}, \underline{1})$ if $\mathbb{1}_P$ is represented in T by $\varphi(\bar{x}, y)$. \square

In 5.3.2(ii) we have already seen that recursively enumerable sets are definable in the standard model. This will now be detailed further:

7.2.3. Theorem. *Let $P \subseteq \omega^n$. Then*

- (i) *P is recursive if and only if P is Δ_1^ω .*
- (ii) *P is recursively enumerable if and only if P is Σ_1^ω .*

Proof. Let T be the theory of $\bar{\omega}$.

(i) \Rightarrow : If P is recursive, then by 7.2.2, P is represented in T by a Σ_1 -formula and by a Π_1 -formula. Since $T \vdash \Omega$, this implies that P is defined by a Σ_1 -formula and by a Π_1 -formula in $\bar{\omega}$ (cf. 5.3.2(i)). Thus P is in Δ_1^ω .

(ii) \Rightarrow : If P is recursively enumerable, then P is the projection of a recursive set P' . We have just seen that P' is defined by a Σ_1 -formula. But then P is defined by a Σ_1 -formula, too.

(ii) \Leftarrow : By 7.1.5 every set in Δ_0^ω is recursive (even primitive recursive). Since P is in Σ_1^ω , P is the projection of a recursive set. Hence P is recursively enumerable (cf. 1.5.3).

(i) \Leftarrow : If P is in Δ_1^ω , then P and its complement are in Σ_1^ω . We have just seen that in this case P and its complement are recursively enumerable. Thus by 1.5.2, P is recursive. \square

7.2.4. Theorem. *If T is a recursive set of \mathcal{L} -sentences and $T \cup \mathbf{PA}$ is consistent, then there is a Π_1 -sentence that is independent of T . Explicitly, the sentence $\beta_{T \cup \mathbf{PA}}$ defined in 5.3.5 is independent of T (it is even independent of $T \cup \mathbf{PA}$) and is provably equivalent modulo $T \cup \mathbf{PA}$ to a Π_1 -sentence.*

Proof. Question 34 of the example sheets. \square

7.3. Kleene's Enumeration theorem.

In 7.2.3 we have seen that every recursively enumerable set is defined by a Σ_1 -formula in the standard model. Surprisingly, all these sets occur as the fibres of a single Σ_1 -formula:

7.3.1. Kleene's Enumeration theorem (S. C. Kleene, pronounced 'KLAY-nee')

There is a Σ_1 -formula $\kappa(x, y)$ (of $\mathcal{L}_{\bar{\omega}}$) in two free variables such that every recursively enumerable subset of ω is of the form

$$\{n \in \omega \mid \bar{\omega} \models \kappa[k, n]\}$$

for some $k \in \omega$.

Proof. Define $P \subseteq \omega^3$ by

$$P(k, n, l) \iff k \text{ is the Gödel number of a } \Sigma_1\text{-formula } \sigma(u) \\ \text{and } l \text{ is the proof number of a proof of } \sigma(\underline{n}) \text{ in } \mathbf{\Omega}.$$

P is recursive since all predicates in its definition are recursive by earlier results (like 4.3 and 5.1.3). By 7.2.3(ii), there is a Σ_1 -formula $\varphi(x, y, z)$ that defines P in $\bar{\omega}$. We take

$$\kappa(x, y) = \exists z \varphi(x, y, z)$$

and show that κ has the required property:

Let $U \subseteq \omega$ be recursively enumerable. By 7.2.3(ii), U is defined in $\bar{\omega}$ by a Σ_1 -formula $\sigma(u)$. Let $k = \ulcorner \sigma \urcorner$. Then for $n \in \omega$ we have

$$\begin{aligned} n \in U &\iff \bar{\omega} \models \sigma[n] \iff \bar{\omega} \models \sigma(\underline{n}) \\ &\iff \mathbf{\Omega} \vdash \sigma(\underline{n}), \text{ by 7.2.1(ii)} \\ &\iff \text{there is } l \in \omega \text{ with } P(k, n, l) \\ &\iff \text{there is } l \in \omega \text{ with } \bar{\omega} \models \varphi[k, n, l] \\ &\iff \bar{\omega} \models \kappa[k, n]. \end{aligned}$$

Hence U is of the desired form. \square

7.3.2. Remarks.

- (i) By 7.2.3(ii) we of course also know that each fibre of $\kappa(x, y)$ in 7.3.1 is recursively enumerable.
- (ii) A corresponding result for recursively enumerable subsets of ω^n follows easily from 7.3.1: Let $f = (f_1, \dots, f_n) : \omega \rightarrow \omega^n$ be a recursive isomorphism (so f is bijective, all f_i and f^{-1} are recursive) and let $\psi(x, y_1, \dots, y_n)$ be the formula

$$\exists z \left(\kappa(x, z) \wedge \text{"}y_1 = f_1(z)\text{"} \wedge \dots \wedge \text{"}y_n = f_n(z)\text{"} \right).$$

Here, " $y_i = f_i(z)$ " stands for a Σ_1 -formula that defines the graph of f_i . Then if $Y \subseteq \omega^n$ is recursively enumerable, also $f^{-1}(Y) \subseteq \omega$ is recursively enumerable, and there is some $k \in \omega$ with $X = \kappa[k, \bar{\omega}]$. Clearly then $Y = f(f^{-1}(Y)) = \psi[k, \bar{\omega}^n]$.

7.3.3. Corollary. The set

$$X = \{n \in \omega \mid \bar{\omega} \models \kappa[n, n]\}$$

is recursively enumerable, but not recursive.

Proof. The set X is recursively enumerable, as it is Σ_1 . By Cantor's Anti-diagonal lemma 5.2.4, the anti-diagonal $\omega \setminus X$ of the relation defined by κ is not a fibre of that relation. Hence by 7.3.1, $\omega \setminus X$ cannot be recursively enumerable, either. Thus X is not recursive. \square

7.4. Hilbert's 10th problem.

The tenth problem in Hilbert's celebrated list of unsolved problems in mathematics from 1900 asks to find a decision procedure for the solvability of polynomial equations over the integers.

In 1970, Yuri Matijasevič, based on the work of many others, mainly M. Davis, H. Putnam and J. Robinson (and certainly starting with Gödel), showed that there is in fact no such decision procedure. In this section we will explain the framework of Hilbert's 10th problem without giving a full proof of its surprising answer.

A **diophantine equation** is an equation of the form

$$P(T_1, \dots, T_n) = Q(T_1, \dots, T_n),$$

where P, Q are polynomials over \mathbb{Z} with non-negative coefficients. One can also talk about diophantine equations over other (semi)-rings and this is studied in the literature. In particular we want to be able to link this question to natural numbers.

7.4.1. Definition. Let R be a commutative unital ring. We say that **Hilbert's 10th problem for R holds** if there is a decision procedure that decides whether diophantine equations have solutions in R . More precisely (and taking the Church-Turing thesis for granted):

The set of Gödel numbers of diophantine equations that have a solution in R is recursive.

In the case $R = \mathbb{Z}$ we want to translate the problem into a question about the standard model $\bar{\omega}$. First some terminology.

7.4.2. Definition. An \exists_1 -**formula** is a formula of $\mathcal{L}_{\bar{\omega}}$ of the form

$$\exists \bar{x} \ t \doteq s,$$

where t and s are $\mathcal{L}_{\bar{\omega}}$ -terms.

An \exists_1^ω -**formula** is a formula of $\mathcal{L}_{\bar{\omega}}$ that is equivalent in $\bar{\omega}$ to a \exists_1 -formula.

A subset of ω^n is called \exists_1^ω if it is defined by a \exists_1^ω -formula (equivalently: by a \exists_1 -formula).

Obviously, $\exists_1^\omega \subseteq \Sigma_1^\omega$. Further, the only feature of Σ_1^ω that is a priori lacking in \exists_1^ω is bounded universal quantification:

7.4.3. Remarks. The following $\mathcal{L}_{\bar{\omega}}$ formulas are universally true in $\bar{\omega}$:

- (i) $x < y \iff \exists z(x + S(z) = y)$
- (ii) $x = y \wedge u = v \iff x^2 + y^2 + u^2 + v^2 = 2xy + 2uv$,
because $x = y \wedge u = v \iff (y-x)^2 + (v-u)^2 = 0 \iff x^2 + y^2 + u^2 + v^2 = 2xy + 2uv$ for all $x, y, u, v \in \omega$.
- (iii) $x = y \vee u = v \iff yv + xu = xv + yu$,
because $x = y \vee u = v \iff (y-x) \cdot (v-u) = 0 \iff yv + xu = xv + yu$ for all $x, y, u, v \in \omega$.

- (iv) $x \neq y \iff x < y \vee y < x$
 (v) $x \not< y \iff y < x \vee y = x$.

Consequently, these equivalences also hold true universally in $\bar{\omega}$ if we replace variables by terms. Thus every quantifier free $\mathcal{L}_{\bar{\omega}}$ -formula is an \exists_1^ω -formula. From (ii) and (iii) (for terms) it also follows that \exists_1^ω is closed under conjunction and disjunction.

Conclusion: The set of \exists_1^ω -formulas contains all quantifier free $\mathcal{L}_{\bar{\omega}}$ -formulas and it is closed under conjunction, disjunction and bounded existential quantification.

7.4.4. Lemma. *The following are equivalent.*

- (i) *Hilbert's 10th problem holds for \mathbb{Z} .*
 (ii) *The set of Gödel numbers of diophantine equations that have a solution in ω is recursive.*

Proof. (i) \Rightarrow (ii). $P(x_1, \dots, x_n) = Q(x_1, \dots, x_n)$ has a solution in ω if and only if $P(u_1^2 + v_1^2 + w_1^2 + z_1^2, \dots, u_n^2 + v_n^2 + w_n^2 + z_n^2) = Q(u_1^2 + v_1^2 + w_1^2 + z_1^2, \dots, u_n^2 + v_n^2 + w_n^2 + z_n^2)$ has as solution in \mathbb{Z} by Lagrange's 4 square theorem.

(ii) \Rightarrow (i) $P(x_1, \dots, x_n) = Q(x_1, \dots, x_n)$ has a solution in \mathbb{Z} if and only if there is some $\varepsilon \in \{-1, 1\}^n$ such that $P(\varepsilon_1 x_1, \dots, \varepsilon_n x_n) = Q(\varepsilon_1 x_1, \dots, \varepsilon_n x_n)$ has a solution in ω . For each $\varepsilon \in \{-1, 1\}^n$ take $P_\varepsilon^+, P_\varepsilon^-, Q_\varepsilon^+, Q_\varepsilon^- \in \mathbb{Z}[x_1, \dots, x_n]$ with non-negative coefficients such that

$$P(\varepsilon_1 x_1, \dots, \varepsilon_n x_n) = P_\varepsilon^+(x_1, \dots, x_n) - P_\varepsilon^-(x_1, \dots, x_n) \text{ and} \\ Q(\varepsilon_1 x_1, \dots, \varepsilon_n x_n) = Q_\varepsilon^+(x_1, \dots, x_n) - Q_\varepsilon^-(x_1, \dots, x_n).$$

Then $P(x_1, \dots, x_n) = Q(x_1, \dots, x_n)$ has a solution in \mathbb{Z} if and only if

$$(*) \quad \bar{\omega} \models \exists \bar{x} \left(\bigvee_{\varepsilon \in \{-1, 1\}^n} P_\varepsilon^+(\bar{x}) + Q_\varepsilon^-(\bar{x}) = Q_\varepsilon^+(\bar{x}) + P_\varepsilon^-(\bar{x}) \right).$$

By 7.4.3 the disjunctions of the term equalities in (*) can be replaced by a single diophantine equation. \square

7.4.5. Remarks.

- (i) As mentioned at the beginning, Matijasevič has shown that Hilbert's 10th problem fails for \mathbb{Z} , we will discuss this below.
 (ii) It is a wide open problem whether Hilbert's 10th problem holds for \mathbb{Q} .
 (iii) Hilbert's 10th problem holds for \mathbb{R} , \mathbb{C} and \mathbb{Q}_p as follows easily from the decidability of these rings.

7.4.6. MRDP-theorem (Matijasevič, Robinson, Davis, Putnam)

$$\exists_1^\omega = \Sigma_1^\omega.$$

Proof. By 7.4.3, the only thing that remains to show is that \exists_1^ω is closed under bounded quantification. This is non-trivial and can be found in [Matija], [DaPuRo]. \square

As a consequence we obtain

7.4.7. Theorem. *Hilbert's 10th problem fails for \mathbb{Z} .*

Proof. By 7.4.4 it suffices to show that the set of Gödel numbers of diophantine equations that have a solution in ω is not recursive. By the MRDP theorem 7.4.6, Kleene's formula $\kappa(x, y)$ (see 7.3.1) is equivalent in $\bar{\omega}$ to a formula

$$\exists \bar{z} P(x, y, \bar{z}) = Q(x, y, \bar{z}),$$

where P, Q are polynomials over \mathbb{Z} with non-negative coefficients. By 7.3.3, the set defined by $\kappa(x, x)$ in $\bar{\omega}$ is not recursive, thus

$$S := \{n \in \omega \mid \bar{\omega} \models \exists \bar{z} P(n, n, \bar{z}) = Q(n, n, \bar{z})\}$$

is not recursive. It follows that the set

$$G = \{\ulcorner P(n, n, \bar{z}) = Q(n, n, \bar{z}) \urcorner \mid n \in \omega \text{ and } P(n, n, \bar{z}) = Q(n, n, \bar{z}) \text{ is solvable in } \omega\}$$

is not recursive either: To see this, note that

$$n \in S \iff \bar{\omega} \models \exists \bar{z} P(n, n, \bar{z}) = Q(n, n, \bar{z}) \iff \ulcorner P(n, n, \bar{z}) = Q(n, n, \bar{z}) \urcorner \in G;$$

Since the function that maps n to $\ulcorner P(n, n, \bar{z}) = Q(n, n, \bar{z}) \urcorner$ is recursive (see section 4), G cannot be recursive.

But then the set of Gödel numbers of diophantine equations that have a solution in ω is not recursive either (notice that the set $\{\ulcorner P(n, n, \bar{z}) = Q(n, n, \bar{z}) \urcorner \mid n \in \omega\}$ is recursive). \square

As a matter of fact there are polynomials over \mathbb{Z} encoding the solvability of all the other polynomials in any number of variables:

7.4.8. Theorem. *There is a polynomial $U(\bar{x}, y) \in \mathbb{Z}[\bar{x}, y]$ (in some number of variables) with the following property:*

For every number of indeterminates T_1, \dots, T_n and all polynomials $F \in \mathbb{Z}[T_1, \dots, T_n]$ one can explicitly construct a number $k_F \in \omega$ such that

$$F(T_1, \dots, T_n) = 0 \text{ is solvable in } \omega \iff U(\bar{x}, k_F) = 0 \text{ is solvable in } \omega.$$

*Each such polynomial U is called a **universal polynomial**.*

Proof. We know that the set G of all Gödel numbers of $\mathcal{L}_{\bar{\omega}}$ -sentences that are provable from Ω is recursively enumerable. By 7.2.3(ii), this set is Σ_1^ω and by the MRDP theorem 7.4.6, there are polynomials $P(\bar{x}, y), Q(\bar{x}, y)$ such that

$$(*) \quad G = \{k \in \omega \mid \bar{\omega} \models \exists \bar{x} P(\bar{x}, \underline{k}) = Q(\bar{x}, \underline{k})\}.$$

We take

$$U(\bar{x}, y) = P(\bar{x}, y) - Q(\bar{x}, y).$$

Now pick any polynomial $F(T_1, \dots, T_n) \in \mathbb{Z}[T_1, \dots, T_n]$ and write it as

$$F^+(T_1, \dots, T_n) - F^-(T_1, \dots, T_n)$$

with polynomials F^+, F^- having only non-negative coefficients. We define

$$k_F = \ulcorner \exists t_1, \dots, t_n F^+(t_1, \dots, t_n) = F^-(t_1, \dots, t_n) \urcorner.$$

Then $F(T_1, \dots, T_n) = 0$ is solvable in $\omega \iff$

$$\begin{aligned} &\iff \bar{\omega} \models \exists t_1 \dots, t_n F^+(t_1, \dots, t_n) = F^-(t_1, \dots, t_n) \\ &\iff \mathbf{\Omega} \vdash \exists t_1 \dots, t_n F^+(t_1, \dots, t_n) = F^-(t_1, \dots, t_n) \text{ by 7.2.1(ii)} \\ &\iff \ulcorner \exists t_1 \dots, t_n F^+(t_1, \dots, t_n) = F^-(t_1, \dots, t_n) \urcorner \in G \text{ by definition of } G \\ &\iff k_F \in G \text{ by choice of } k_F \\ &\iff \bar{\omega} \models \exists \bar{x} P(\bar{x}, \underline{k}_F) = Q(\bar{x}, \underline{k}_F) \text{ by } (*). \end{aligned}$$

□

The existence of universal polynomials has remarkable consequences. First recall from 7.2.4 that for any consistent and recursive strengthening T of Peano arithmetic, the sentence β_T that is independent of T is in $\Pi_1(T)$. Further it is possible to construct β_T explicitly.

Suppose now that $\bar{\omega} \models T$. Then $\bar{\omega} \models \beta_T$ (we have $\neg(n \text{ IsProofOf}_T \ulcorner \beta_T \urcorner$), hence $T \vdash \neg \text{IsProofOf}_T(n, \ulcorner \beta_T \urcorner)$ for all $n \in \omega$, implying $\bar{\omega} \models \beta_T$).

Then the meaning of β_T in ω is the following: By Matijasiewiĉ, β_T says that a certain explicitly constructible diophantine equation has no solution in ω . Now 7.4.8 says we can explicitly compute a natural number k_T such that the truth of β_T in $\bar{\omega}$ is equivalent to the unsolvability of $U(\bar{x}, k_T) = 0$ in ω .

In this sense, a single polynomial $U(\bar{x}, y)$ captures the incompleteness of all recursive subsets of $\text{Th}(\bar{\omega})$.

We'll come back to another consequence of 7.4.8 shortly, when we have Gödel's second incompleteness theorem.

8. GÖDEL'S SECOND INCOMPLETENESS THEOREM

This section is not examinable.

Throughout this section, \mathcal{L} denotes the language $\mathcal{L}_{\bar{\omega}}$ of arithmetic and T denotes a consistent and recursive set of \mathcal{L} -sentences with $T \vdash PA$. Then the binary predicate IsProofOf_T is recursive and by 7.2.2 there is a Σ_1 -formula $\underline{\text{IsProofOf}}_T(u, x)$ representing IsProofOf_T in T . We define $\Box_T(x)$ to be the Σ_1 -formula

$$\exists u \underline{\text{IsProofOf}}_T(u, x).$$

As long as T is fixed we drop the index T and just write $\Box(x)$.

So, $\Box(x)$ is a Σ_1 -formula such that for every $\mathcal{L}_{\bar{\omega}}$ -sentence φ we have

$$\bar{\omega} \models \Box(\ulcorner \varphi \urcorner) \iff T \vdash \varphi$$

Now choose a Σ_1 -formula that represents the predicate $z = \prec x, y \succ$ in T and write this formula as $\underline{z = \prec x, y \succ}$.

Let $\nabla(x)$ be the formula

$$\exists y \left(\underline{y = \prec \ulcorner \neg \urcorner, x \succ} \wedge \Box(y) \right)$$

So, $\nabla(x)$ is a Σ_1 -formula such that for every $\mathcal{L}_{\bar{\omega}}$ -sentence φ we have

$$T \vdash \nabla(\ulcorner \varphi \urcorner) \iff \Box(\ulcorner \neg \varphi \urcorner)$$

and

$$\bar{\omega} \models \nabla(\ulcorner \varphi \urcorner) \iff T \vdash \neg \varphi.$$

Finally we define an $\mathcal{L}_{\bar{\omega}}$ -sentence Con_T as

$$\neg \exists x \left(\Box(x) \wedge \nabla(x) \right).$$

Thus Con_T is a Π_1 -sentence and

$$\bar{\omega} \models \text{Con}_T \iff T \text{ is consistent}$$

8.1. Gödel's second incompleteness theorem

If T is a recursive set of $\mathcal{L}_{\bar{\omega}}$ -sentences with $T \vdash \mathbf{PA}$ (as always in this section), then

$$T \text{ is consistent} \implies T \not\vdash \text{Con}_T.$$

Before proving this, let us go back and consider the impact of the existence of universal polynomials again. Let $U(\bar{x}, y)$ be a universal polynomial as constructed in 7.4.8.

Given any recursive strengthening T of \mathbf{PA} , we can explicitly construct a Π_1 -sentence Con_T , which asserts the consistency of T and which is not provable in T , unless T is inconsistent. By Matijasevič and 7.4.8, we can explicitly construct a natural number k_T such that the truth of Con_T in $\bar{\omega}$ is equivalent to the unsolvability of $U(\bar{x}, k_T) = 0$ in ω .

Hence if we believe that T is consistent, then we must also believe that $U(\bar{x}, k_T) = 0$ is unsolvable in ω . In this sense, a single polynomial $U(\bar{x}, y)$ captures the consistency assertions of every recursive strengthening of Peano Arithmetic.

For the proof of 8.1 we need some preparations.

8.2. Lemma. *Let φ be an \mathcal{L} -sentence.*

(i) *If $T \vdash \varphi$, then $T \vdash \Box(\ulcorner \varphi \urcorner)$*

(ii) *If T is ω -consistent (e.g. if $\bar{\omega} \models T$, see question 26), then*

$$T \vdash \varphi \iff T \vdash \Box(\ulcorner \varphi \urcorner)$$

Proof. (i) As $T \vdash \varphi$, there is some $n \in \omega$ with $n \text{ IsProofOf}_T \ulcorner \varphi \urcorner$ (take n to be the proof number of a proof of φ in T). Since IsProofOf_T represents IsProofOf_T in T we know $T \vdash \text{IsProofOf}_T(n, \ulcorner \varphi \urcorner)$. In particular $T \vdash \exists x \text{ IsProofOf}_T(x, \ulcorner \varphi \urcorner)$.

(ii) Now suppose $T \not\vdash \varphi$. Then for each $n \in \omega$ we have

$$\neg \left(n \text{ IsProofOf}_T \ulcorner \varphi \urcorner \right).$$

Since IsProofOf_T represents IsProofOf_T in T we know

$$T \vdash \neg \text{IsProofOf}_T(n, \ulcorner \varphi \urcorner) \text{ for all } n \in \omega.$$

As T is ω -consistent, we get $T \not\vdash \exists x \text{ IsProofOf}_T(x, \ulcorner \varphi \urcorner)$, as required. \square

It should be mentioned that the implication in (i) of 8.2 cannot be reversed (unless T is ω -consistent). See question 26 of the example sheets for the notion of ω -consistency and a discussion of it.

8.3. Theorem. *(Internalization)*

If φ is a Σ_1 -sentence and

$$T \vdash \varphi \leftrightarrow \Box(\ulcorner \psi \urcorner)$$

for some $\mathcal{L}_{\bar{\omega}}$ -sentence ψ , then

$$T \vdash \varphi \rightarrow \Box(\ulcorner \varphi \urcorner).$$

Proof. By 7.2.1(ii) we know that

$$\bar{\omega} \models \varphi \Rightarrow T \vdash \varphi$$

and so by definition of $\Box(x)$, also

$$(*) \quad \bar{\omega} \models \varphi \rightarrow \Box(\ulcorner \varphi \urcorner)$$

Now, using the assumption that $T \vdash \varphi \leftrightarrow \Box(\ulcorner \psi \urcorner)$ one can *internalise* the proof of (*) and show that (*) actually follows with the aid of the induction principle formulated in **PA**. Internalization here essentially means that the coding process can actually be proved in Peano arithmetic; e.g. we know that the coding process from section 4 can be done entirely with primitive recursive functions. Then one shows that every primitive recursive function $F : \omega^n \rightarrow \omega$ is *provably recursive* meaning that there is a Σ_1 -formula $\gamma(x_1, \dots, x_n, y)$ such that

$$\text{PA} \vdash \forall \bar{x} \exists! y \gamma(\bar{x}, y) \text{ and } \text{PA} \vdash \gamma(\bar{a}, F(\bar{a})) \text{ for all } \bar{a} \in \omega^n.$$

This is a tedious process which is skipped here. Instead we refer to [Rautenberg, section 7.1]. \square

8.4. Proof of 8.1

From the proof of 7.2.4 we know that we may choose $\neg\beta_T$ as a Σ_1 -sentence. Now recall that

$$\Omega \vdash \beta_T \longleftrightarrow \forall u \neg (\text{IsProofOf}_T(u, \ulcorner \beta_T \urcorner)),$$

so

$$\Omega \vdash \neg\beta_T \longleftrightarrow \exists u \text{IsProofOf}_T(u, \ulcorner \beta_T \urcorner),$$

i.e.

$$(*) \quad \Omega \vdash \neg\beta_T \longleftrightarrow \Box(\ulcorner \beta_T \urcorner)$$

and so

$$(**) \quad \Omega \vdash \neg\beta_T \longleftrightarrow \nabla(\ulcorner \neg\beta_T \urcorner)$$

By (*) and 8.3 we know that

$$(+)\quad T \vdash \neg\beta_T \longrightarrow \Box(\ulcorner \neg\beta_T \urcorner).$$

Now (+) and (**) imply

$$(\dagger)\quad T \vdash \neg\beta_T \longrightarrow \Box(\ulcorner \neg\beta_T \urcorner) \wedge \nabla(\ulcorner \neg\beta_T \urcorner).$$

in particular

$$T \vdash \neg\beta_T \longrightarrow \neg\text{Con}_T.$$

(Recall that Con_T is $\neg\exists x(\Box(x) \wedge \nabla(x))$). Thus

$$T \vdash \text{Con}_T \longrightarrow \beta_T$$

and as β_T is independent of T , we cannot have $T \vdash \text{Con}_T$. \square

As was outlined by Gödel, the statement of the second incompleteness theorem can itself be proved inside T . This is also achieved via internalization, i.e. one can show

$$T \vdash \text{Con}_T \longrightarrow \neg\Box_T(\ulcorner \text{Con}_T \urcorner).$$

Note that this does not contradict 8.3, since Con_T is not Σ_1 ; again, for reference see [Rautenberg, section 7], which in addition gives several variants and generalizations of Gödel's second incompleteness theorem (e.g. ZFC is addressed, which in fact is easier to handle than PA, because the internalization process is easier)

REFERENCES

- [BCSS] Lenore Blum, Felipe Cucker, Michael Shub, and Steve Smale. Complexity and real computation. Springer-Verlag, New York, 1998. With a foreword by Richard M. Karp.
- [Cutland] Nigel Cutland. Computability. Cambridge University Press, Cambridge, 1980. An introduction to recursive function theory.
- [DaPuRo] Martin Davis, Hilary Putnam, and Julia Robinson. The decision problem for exponential diophantine equations. Ann. of Math. (2), 74:425–436, 1961. 79
- [Goedel63] Kurt Gödel. On formally undecidable propositions of *Principia Mathematica* and related systems. Translated by B. Meltzer, with an introduction by R. B. Braithwaite. Basic Books Inc. Publishers, New York, 1963.
- [Goedel31] Kurt Gödel. Über formal unentscheidbare Sätze der *principia mathematica* und verwandter Systeme. I. Monatsh. Math., 149(1):1–30, 2006. Reprinted from Monatsh. Math. Phys. 38 (1931), 173–198 [MR1549910], With an introduction by Sy-David Friedman. 57
- [Grzegorzcyk] Andrzej Grzegorzcyk. Undecidability of some topological theories. Fund. Math., 38:137–152, 1951. 71
- [Matija] Ju. V. Matijasevič. The Diophantineness of enumerable sets. Dokl. Akad. Nauk SSSR, 191:279–282, 1970. 79
- [Muraws1999] Roman Murawski. Recursive functions and metamathematics, volume 286 of Synthese Library. Kluwer Academic Publishers Group, Dordrecht, 1999. Problems of completeness and decidability, Gödel's theorems.
- [Odifreddi] Piergiorgio Odifreddi. Classical recursion theory, volume 125 of Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Co., Amsterdam, 1989. The theory of functions and sets of natural numbers, With a foreword by G. E. Sacks.
- [Putnam] Hilary Putnam. Decidability and essential undecidability. J. Symb. Logic, 22:39–54, 1957.
- [Rautenberg] Wolfgang Rautenberg. A concise introduction to mathematical logic. Universitext. Springer, New York, second edition, 2006. With a foreword by Lev Beklemishev. 83, 84
- [Robinson] Julia Robinson. The undecidability of algebraic rings and fields. Proc. Amer. Math. Soc., 10:950–957, 1959. 71
- [Smullyan] Raymond M. Smullyan. Gödel's incompleteness theorems, volume 19 of Oxford Logic Guides. The Clarendon Press Oxford University Press, New York, 1992.
- [Tarski] Alfred Tarski. Undecidable theories. Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Company, Amsterdam, 1953. In collaboration with Andrzej Mostowski and Raphael M. Robinson.
- [PrLog] Marcus Tressl. Predicate logic, 2016. Lecture Notes <http://personalpages.manchester.ac.uk/staff/Marcus.Tressl/teaching/Goedel/PredicateLogic.pdf>.
- [Ziegler] Martin Ziegler. Mathematische Logik. Mathematik Kompakt. [Compact Mathematics]. Birkhäuser Verlag, Basel, 2010.

INDEX

- $\mathbb{N} = \{1, 2, 3, \dots\}$,
- $\mathbb{N}_0 = \{0, 1, 2, 3, \dots\}$,
- $\mathcal{P}(X)$ = power set of X ,
- $\text{Maps}(X, Y)$ = set of all maps $X \rightarrow Y$,
- (L, R) , 7
- $(a)_i$, 15
- A_R , 6
- E_R , 6
- $F^{\mathcal{M}}$, 31
- I_i^n , 1
- P^Σ , 56
- $R^{\mathcal{M}}$, 31
- R is represented by φ in Σ , 39
- $[l]$ for a letter l , 47
- $\square_T(x)$, 82
- Δ_0 -formula, 72
- $\Delta_0(T)$, 73
- Δ_0^ω , 73
- Δ_m^ω , 73
- ΠF , 12
- Π_1 -formula, 72
- Π_m , 72
- $\Pi_m(T)$, 73
- Π_m^ω , 73
- Σ -representable, 39
- Σ -represented, 39
- ΣF , 12
- Σ_1 -formula, 72
- Σ_m , 72
- $\Sigma_m(T)$, 73
- Σ_m^ω , 73
- β -function, 9
- ℓ , 15
- \exists_1 -formula, 78
- \exists_1^ω -formula, 78
- $\bigvee_{i=1}^n \varphi_i$, 28
- $\bigwedge_{i=1}^n \varphi_i$, 28
- Seq**, 16
- Ω , 41
- $\mathbb{1}_R$, 1
- $\mathbb{N} = \{1, 2, 3, \dots\}$, 1
- $\text{Fml}(\mathcal{L})$ or $\text{Fml}(\mathcal{L})$, 27
- $\text{Fml}_k(\mathcal{L})$, 27
- $\text{Fr}(\varphi)$, 29
- $\text{Fr}(t)$, 29
- Num**, 56
- Proof** $_\Sigma$, 52
- $\text{Sen}(\mathcal{L})$, 29
- $\text{Th}(\mathcal{M})$, 53
- $\text{card}(\mathcal{L})$, 27
- $\text{tm}(\mathcal{L})$ or $\text{tm}(\mathcal{L})$, 25
- $\text{tm}_k(\mathcal{L})$, 25
- IsProofOf** $_\Sigma$, 62
- $\text{Inf}F$, 12
- $\text{Pair}(x, y)$, 7
- $\text{Sup}F$, 12
- \mathcal{L} -formula, 27
- \mathcal{L} -terms, 25
- $\mathcal{L} \subseteq \mathcal{L}'$, 25
- $\mathcal{L}_{\bar{\omega}}$, 41
- \mathcal{M} satisfies $\varphi(a_1, \dots, a_n)$, 32
- \models , 32
- $\Sigma \models \Phi$, 32
- $\mathcal{M} \models \Sigma(\bar{a})$, 32
- $\mathcal{M} \models \varphi(a_1, \dots, a_n)$, 32
- μx , 1
- $\nabla(x)$, 82
- $\omega = \{0, 1, 2, 3, \dots\}$, 1
- φ represents F in Σ , 39
- φ represents R in Σ , 39
- $\varphi(a_1, \dots, a_n)$ holds in \mathcal{M} , 32
- $\varphi(x_1, \dots, x_n) \in \text{Fml}(\mathcal{L})$, 30
- $\varphi(x_1/t_1, \dots, x_n/t_n)$, 30
- $\varphi[\mathcal{M}^n]$, 60
- $\prec a_1, \dots, a_n \succ$, 15
- $\lceil \Sigma \rceil$, $\lceil T \rceil$, etc., 48
- $\lceil \varphi \rceil$, 47
- $\lceil t \rceil$, 47
- \underline{a} , 39
- \underline{n} , 39
- IsProofOf** $_\Sigma(x, y)$, 62
- \vdash , 33
- $\vdash \mathcal{L}$, 33
- $a \upharpoonright_i$, 16
- $a \upharpoonleft_i$, 16
- $c(\varphi)$, 28
- $c(t)$, 26
- $c^{\mathcal{M}}$, 31
- n -ary, 24
- $t(x_1, \dots, x_n) \in \text{tm}(\mathcal{L})$, 30
- $t(x_1/t_1, \dots, x_n/t_n)$, 30
- $t^{\mathcal{M}}(a_1, \dots, a_n)$, 31
- x is free in φ for y , 30
- y is substitutable for x in φ , 30
- $\text{at-Fml}(\mathcal{L})$, 26
- $|\mathcal{M}|$, 31
- \overline{F} , 16
- $a \hat{=} b$, 16
- PA**, 57
- Ackermann function, 18
- almost subtraction, 5
- alphabet, 24
 - countable, 24
 - finite, 24
 - infinite, 24
 - uncountable, 24
- anti-diagonal, 57
- arithmetic sets, 60
- arity, 24
 - of a function symbol, 24
 - of a predicate symbol, 24

- of a relation symbol, 24
- atomic \mathcal{L} -formula, 26
- Axiom
 - logical, 33
- axiom system of T , 53
- bound occurrence, 29
- bounded μ -recursion, 6
- bounded quantification, 6
- Cantor's Anti-diagonal lemma, 57
- cardinality of an alphabet of a language, 25
- carrier of a structure, 31
- Church's theorem, 57
- closed term, 29
- code
 - of a formula, 47
 - of a term, 47
- cofinite, 4
- collection schema, 74
- Compactness Theorem, 36
- complete theory, 53
- Completeness Theorem, 35
- complexity
 - of an \mathcal{L} -formula, 28
 - of an \mathcal{L} -term, 26
- computable
 - function, 2
- computably enumerable
 - set, 22
- concatenation function, 16
- consistent, 34
- constant symbol, 24
- constant term, 29
- Converse of the representability theorem, 58
- coordinate function, 1, 15
- countable
 - alphabet, 24
 - language, 27
 - structure, 31
- decidable
 - set of sentences, 53
 - structure, 68
 - theory, 53
- deduction, 33
- deductive closure, 53
- deductively closed, 53
- definable
 - sets definable in \mathcal{M} , 60
 - structures definable in \mathcal{M} , 65
- definition of \mathcal{L}^+ in \mathcal{L} , 65
- diophantine equation, 78
- domain of a structure, 31
- embedding, 42
- equality symbol, 24
- extension of languages, 25
- finite
 - alphabet, 24
 - language, 27
 - structure, 31
- Fixed Point Lemma, 60
- formal proof, 33
 - of φ , 34
- formula, 27
 - atomic, 26
- free
 - x is free in φ for t , 30
 - occurrence, 29
 - variable, 29
- function symbol, 24
- Gödel number, 47
- Gödel's β -function, 9
- Gödel's First Incompleteness Theorem, 57
- Gödel's second incompleteness theorem, 82
- Hilbert's 10th problem for R holds, 78
- homomorphism, 41
- inconsistent, 34
- independent of Σ , 62
- induction statements, 57
- infinite
 - alphabet, 24
 - language, 27
 - structure, 31
- interpretable in \mathcal{M} , 65
- interpretation
 - definition of a structure in another structure, 65
 - dimension of an interpretation, 65
 - of a language in another language, 65
 - of a structure in another structure, 65
 - of constant symbols, 31
 - of function symbols, 31
 - of relation symbols, 31
 - universe of an interpretation, 65
- Kleene's Enumeration theorem, 77
- language, 27
 - countable, 27
 - finite, 27
 - infinite, 27
 - uncountable, 27
- language extension, 25
- Lemma of self-reference, 61
- length function, 15
- letter, 24
- logical Axioms, 33
- logical symbols, 24
- logically implies, 32
- Matijasevič, MRDP theorem, 79
- model
 - at (a_1, \dots, a_n) , 32

- Modus Ponens, 33
- MRDP-theorem, 79
- numerical language, 39
- Pairing Function, 7
- Peano Arithmetic, 57
- predicate symbol, 24
- prenex normal form, 36
- Prenex Normal Form Theorem, 37
- primitive recursion from H with initial value h , 11
- primitive recursive
 - function, 11
 - relation, 11
 - set, 11
- proof, 33
 - of φ , 34
- proof number, 52
- proves, 33
- quantifier free, 27
- quantifiers, 28
- Recursion on previous values, 17
- recursive
 - function, 2
 - interpretation, 68
 - relation, 2
 - set, 2
 - set of formulas, 52
- recursively axiomatizable, 53
- recursively enumerable
 - set, 22
 - set of formulas, 52
- relation symbol, 24
- Representability Theorem, 43
- represented in Σ , 39
- restriction function, 16
- Robinson Arithmetic, 41
- satisfiable, 32
- scope, 29
- sentence, 29
- sequence number, 15
- set of variables, 24
- signature, 27
- similarity type, 25
- Soundness Theorem, 35
- standard model, 41
- strongly undecidable, 68
- structure, 31
 - carrier of, 31
 - countable, 31
 - domain of, 31
 - finite, 31
 - infinite, 31
 - size of, 31
 - uncountable, 31
 - universe of, 31
- subformula, 28
- substitutable, 30
- substitution, 30
- symbol, 24
- symbol number, 47
- Tarski's theorem on strongly undecidable structures, 69
- Tarski's Undefinability of Truth, 62
- term, 25
 - closed, 29
 - constant, 29
- term representing a function, 39
- Theorems
 - Unique Readability Theorem
 - for formulas, 27
 - for terms, 26
- theorems
 - Cantor's Anti-diagonal lemma, 57
 - Church's theorem, 57
 - Compactness Theorem, 36
 - Completeness Theorem, 35
 - Converse of the representability theorem, 58
 - Fixed Point Lemma, 60
 - Gödel's First Incompleteness Theorem, 57
 - Gödel's second incompleteness theorem, 82
 - Kleene's Enumeration theorem, 77
 - MRDP-theorem, 79
 - Negation Theorem, 22
 - Prenex Normal Form Theorem, 37
 - Recursion on previous values, 17
 - Representability Theorem, 43
 - Soundness Theorem, 35
 - Tarski's theorem on strongly undecidable structures, 69
 - Tarski's Undefinability of Truth, 62
- theory, 53
 - complete, 53
 - of a structure, 53
- uncountable
 - alphabet, 24
 - language, 27
 - structure, 31
- undecidable
 - set of sentences, 53
 - strongly, 68
 - structure, 68
 - theory, 53
- undecidable in Σ , 62
- Unique Readability Theorem
 - for formulas, 27
 - for terms, 26
- universal polynomial, 80
- universe of a structure, 31

variable, 24
 bound occurrence, 29
 free, 29
 free occurrence, 29

THE UNIVERSITY OF MANCHESTER, SCHOOL OF MATHEMATICS, OXFORD ROAD, MANCHESTER
M13 9PL, UK

HOME PAGE: [HTTP://PERSONALPAGES.MANCHESTER.AC.UK/STAFF/MARCUS.TRESSL/INDEX.PHP](http://personalpages.manchester.ac.uk/staff/marcus.tressl/index.php)

Email address: `marcus.tressl@manchester.ac.uk`