Mathematische Logik - WS14/15

Iosif Petrakis
petrakis@mathematik.uni-muenchen.de

January 14, 2015

These notes include part of the material discussed in the Tutorial and in the Exercises that correspond to the lecture course "Logik" of Prof. Dr. Hans-Dieter Donder. Of course, possible mistakes in these notes are not related to Prof. Donder at all. Many extra, or optional exercises can be found here.

Please feel free to send me your comments, or your suggestions regarding these notes.

The sign ★ denotes an interesting and non-trivial exercise. Usually such an exercise is optional.

1. Inductive Definitions

The most characteristic example of an inductive (or recursive) definition is that of a *natural number*. It can be given using the following two rules

$$\frac{}{0 \in \mathbb{N}} \qquad \frac{n \in \mathbb{N}}{S(n) \in \mathbb{N}},$$

where $S(n)$ denotes the successor of n. Note though, that this definition alone does not determine a unique set; for example the rationals $\mathbb{Q}$ or the reals $\mathbb{R}$ satisfy the same rules. We determine $\mathbb{N}$ by postulating that $\mathbb{N}$ is the least set satisfying the above rules. We do so by stating the following induction axiom:

$$A(0) \to \forall_n(A(n) \to A(S(n))) \to \forall_n(A(n)),$$

where A is a formula representing a predicate on natural numbers. If A is any formula, then the above principle is the well-known full induction principle on natural numbers (if we restrict the range of A's we get other weaker induction principles). Its interpretation is the following: Suppose that A satisfies the two rules, $A(0)$ and $\forall_n(A(n) \to A(S(n)))$, i.e., as it is usually said, A is a "competitor" predicate to $\mathbb{N}$, then, if we view A as the set of all objects such that $A(n)$, $\mathbb{N} \subseteq A$, i.e., $\forall_n(A(n))$, or more precisely, $\forall_x(\mathbb{N}(x) \to A(x))$. A consequence of this inductive characterization of $\mathbb{N}$ is that if we want to define a function $f : \mathbb{N} \to X$, where X is a set, it suffices to define it on 0, and provide a rule G which gives the value of f on $S(n)$ through the value of f on n. I.e., we have the following proposition:

Proposition 1. *If X is a set, $x_0 \in X$ and $G : X \to X$ is a function, then there exists a unique function $f : \mathbb{N} \to X$ such that $f(0) = x_0$ and $f(S(n)) = G(f(n))$.*

Proof. It is direct to see that the above defined f is a function with domain $\mathbb{N}$ and range in X. To show its uniqueness we suppose that there exists $g : \mathbb{N} \to X$ satisfying the above two conditions, and we apply the induction principle on $A(n) := (f(n) = g(n))$ to show that $\forall_n(A(n))$. $\square$

So, we cannot see an inductive definition without its corresponding induction principle. If we consider now the inductive definition of a *formula* in Classical Propositional Calculus

$$\frac{p \in P}{p \in \text{Form}} \qquad \frac{\phi \in \text{Form}}{\neg\phi \in \text{Form}} \qquad \frac{\phi, \psi \in \text{Form}}{\phi \vee \psi \in \text{Form}},$$

the corresponding full induction principle is

$$A(p) \to \forall_\phi(A(\phi) \to A(\neg\phi)) \to \forall_{\phi,\psi}(A(\phi) \to A(\psi) \to A(\phi \vee \psi)) \to \forall_\phi(A(\phi)),$$

where A is any formula of our meta-language, for example this could be the language of set theory. Some times in the bibliography one can find one more clause in the above inductive definition of the form "there are no other formulas except the ones determined by the previous rules", and then the induction principle is proved as a theorem. It is clear though, that the added rule is a disguised form of the induction principle. Since this added rule is not also a formal statement, we propose in these notes to understand an inductive definition as a pair

$$(\text{Inductive Rules}, \ \text{Induction Principle}).$$

Again, a consequence of this inductive characterization of Form is that if we want to define a function $F : \text{Form} \to X$, where X is a set, it suffices to define it on the prime formulas P, and then provide firstly a rule $G_\neg$ which gives the value of F on $\neg\phi$ through the value of F on ϕ, and secondly a rule $G_\vee$ which gives the value of F on $\phi \vee \psi$ through the value of F on ϕ and ψ. I.e., we have the following proposition:

Proposition 2. *If X is a set, $f : P \to X$, $G_\neg : X \to X$ and $G_\vee : X \times X \to X$ are given functions, then there exists a unique function $F : \text{Form} \to X$ satisfying*

$$F(p) = f(p), \ \textit{for each } p,$$

$$F(\neg\phi) = G_\neg(F(\phi)),$$

$$F(\phi \vee \psi) = G_\vee(F(\phi) \vee F(\psi)).$$

Proof. **Exercise 1**. □

Consider the function $V : \text{Form} \to \mathcal{P}(P)$, where $\mathcal{P}(P)$ denotes the power set of P, and $V(\phi)$ is the set of propositional variables occurring in ϕ defined by

$$V(p) := \{p\},$$

$$V(\neg\phi) := V(\phi),$$

$$V(\phi \vee \psi) := V(\phi) \cup V(\psi).$$

The unique existence of V is guaranteed by Proposition 2.

Exercise 2: Which are the functions $f, G_\neg, G_\vee$ that correspond to the function V?

The set $2 = \{0, 1\}$ (in the lecture course it is also written as $\{w, f\}$, or you can see it elsewhere as $\{\mathtt{tt}, \mathtt{ff}\}$) is the simplest *boolean algebra* i.e., a complemented distributive lattice, or a commutative ring with 1 in which every element is idempotent ($p^2 = p$). These algebraic structures are very important in mathematical logic and topology. So, we define on 2 the following operations:

$$\neg 0 = 1, \quad \neg 1 = 0, \quad 0 \vee b = b, \quad 1 \vee b = 1,$$

for each $b \in 2$. Also the ring operations are defined from $\neg, \vee$ as follows:

$$b \cdot c := b \wedge q,$$

$$b + c := (b \wedge \neg c) \vee (\neg b \wedge c),$$

where $b \wedge c := \neg b \vee c$.

We have now all the tools to understand how a *truth valuation* $W : \mathrm{Form} \to 2$ works. If $W : P \to 2$, $G_\neg = \neg$, and $G_\vee = \vee$ are given, then by Proposition 2 there exists a unique function $W^* : \mathrm{Form} \to 2$ satisfying:

$$W^*(p) = W(p), \text{ for each } p,$$

$$W^*(\neg\phi) = \neg W^*(\phi),$$
$$W^*(\phi \vee \psi) = W^*(\phi) \vee W^*(\psi),$$

where the $\vee$ on the left side of the last equality is the logical connective, and the $\vee$ on the right side is the boolean operation.

Exercise 3: Give an example of a function $F : \mathrm{Form} \to 2$ which is not an extension of a truth valuation.

We are in position now to fully grasp the formulation of the following proposition given in the lecture course.

Proposition 3. *If $W_1, W_2 : P \to 2$ are truth valuations, then*

$$\forall_\phi (W_{1\restriction V(\phi)} = W_{2\restriction V(\phi)} \to W_1^*(\phi) = W_2^*(\phi)).$$

Proof. **Exercise 4**.
Hint: Apply the induction principle corresponding to the inductive definition of formulas on

$$A(\phi) := W_{1\restriction V(\phi)} = W_{2\restriction V(\phi)} \to W_1^*(\phi) = W_2^*(\phi),$$

using the definition of $V(\phi)$. $\square$

Exercise 5: (i) Given the field structure $(\mathbb{R}, +, \cdot, 0, 1)$ of the real numbers, define inductively the set of rationals $\mathbb{Q}$.
(ii) Which is the corresponding induction principle?

2. Classical Propositional Calculus

If $\phi \in \mathrm{Form}$, then it is called a *tautology*, if

$$\forall_{W \in 2^P}(W^*(\phi) = 1),$$

where X^Y denotes the set of all functions $f : Y \to X$. A formula ϕ is called a *contradiction*, if

$$\forall_{W \in 2^P}(W^*(\phi) = 0).$$

Exercise 6: (i) Give an example of a tautology, and an example of a contradiction.
(ii) Show that ϕ is a tautology iff $\neg\phi$ is a contradiction.

(iii) Explain why Proposition 3 guarantees that there is a (semantic) algorithm deciding if a formula ϕ is a tautology or not. Although the notion of a Yes/No-algorithm is not yet formally defined, what we mean by it is a recipe which can be executed in a finite amount of time and provides effectively a Yes or No-answer to a given question.
(iv) Describe explicitly the above algorithm for a specific formula ϕ of your choice.
(v) After you learn the definition of a tautology in Classical Predicate Calculus, try to guess if there is a similar (semantic) algorithm deciding if a formula in Predicate Calculus is a tautology or not.

Proposition 4. *If $n \in \mathbb{N}$, $M_n = 2^{\{p_0,\ldots,p_n\}}$ and $F : M_n \to 2$, then*

$$\exists_\phi (V(\phi) \subseteq \{p_0, \ldots, p_n\} \ \wedge \ \forall_W (W^*(\phi) = F(W_{\upharpoonright \{p_0,\ldots,p_n\}})).$$

The fact that the above proposition expresses the *completeness of the connectives* $\{\neg, \vee\}$, that is the sufficiency of $\{\neg, \vee\}$ in writing equivalent forms to all formulas, is related to the general definition of a connective. Maybe more will be added on that later.

Exercise 7. Show that the set of connectives $\{\wedge, \to\}$ is not complete.

Hint: Since $\{\neg, \vee\}$ is a complete set of connectives, the set $\{\neg, \wedge\}$ is also complete (why?). Then try to show that $\neg$ is not expressible within $\{\wedge, \to\}$. If $\{\wedge, \to\}$ was complete, there would exist some formula ϕ including $p, \wedge, \to$ and equivalent to $\neg p$ i.e., $\forall_W (W^*(\phi) = W^*(\neg p))$, for some fixed $p \in P$. Show that this cannot happen.

3. The basic definitions of Classical Predicate Calculus

Exercise 8. Write down the Induction Principle that corresponds to the inductive definition of L-terms[1]

$$\frac{v \in \mathrm{Var}}{v \in \mathrm{Term}}, \quad \frac{c \in \mathrm{Konst}}{c \in \mathrm{Term}}, \quad \frac{t_1, \ldots, t_n \in \mathrm{Term}, \ f \in \mathrm{Funk}^n}{f(t_1, \ldots, t_n) \in \mathrm{Term}},$$

where Funk^n denotes the set of function symbols of L with arity n. Formulate the theorem of recursive definition on Term which corresponds to Propositions 1 or 2.

Exercise 9. Write down the Induction Principle that corresponds to the inductive definition of L-formulas

$$\frac{t_1, t_2 \in \mathrm{Term}}{t_1 = t_2 \in \mathrm{Form}}, \quad \frac{t_1, \ldots, t_n \in \mathrm{Term}, \ R \in \mathrm{Rel}^n}{R(t_1, \ldots, t_n) \in \mathrm{Form}},$$

[1] For simplicity we avoid the subscript L from the symbols $\mathrm{Var}, \mathrm{Konst}, \mathrm{Term}, \mathrm{Form}$.

$$\frac{\phi \in \text{Form}}{\neg\phi \in \text{Form}}, \quad \frac{\phi, \psi \in \text{Form}}{\phi \vee \psi \in \text{Form}}, \quad \frac{\phi \in \text{Form},\ x \in \text{Var}}{\exists_x \phi \in \text{Form}},$$

where Rel^n denotes the set of relation symbols of L with arity n. Formulate the theorem of recursive definition on Term which corresponds to Propositions 1 or 2. For that it will be helpful to write the last inductive rule as follows:

$$\frac{\phi \in \text{Form},\ x_i \in \text{Var}}{\exists_{x_i} \phi \in \text{Form}},$$

where $\text{Var} = \{x_n \mid n \in \mathbb{N}\}$.

4. The definition of L-sentences

As it is said in [2] p.75, the sentences are usually the most interesting formulas. The others lead a second-class existence; they are used primarily as building blocks for sentences.
First we define recursively the trivial concept "*the set of variables occurring in the term t*", $V(t)$, by the clauses:

$$V(x) = \{x\}$$
$$V(c) = \emptyset$$
$$V(ft_1, \ldots, t_n) = \bigcup_{i=1}^{n} V(t_i).$$

Next we define recursively the trivial concept "*the set of variables occurring in the formula ϕ*", $V(\phi)$, by the clauses:

$$V(t_1 = t_2) = V(t_1) \cup V(t_2)$$
$$V(Rt_1, \ldots, t_n) = \bigcup_{i=1}^{n} V(t_i)$$
$$V(\neg\phi) = V(\phi)$$
$$V(\exists_x \phi) = V(\phi) \cup \{x\}.$$

We define next intuitively the concept "*the variable x occurs free in the formula ϕ*", in symbols $\text{free}_x(\phi)$, recursively as follows (p denotes an atomic formula):

$\text{free}_x(p) \leftrightarrow x$ occurs in p.
$\text{free}_x(\neg\phi) \leftrightarrow \text{free}_x(\phi)$.
$\text{free}_x(\phi \vee \psi) \leftrightarrow \text{free}_x(\phi)$ or $\text{free}_x(\psi)$.
$\text{free}_x(\exists_{x_i}\phi) \leftrightarrow x \neq x_i$ and $\text{free}_x(\phi)$.

Formally we define the same object as a function

$$\text{free}_x : \text{Form}_L \to 2$$

$$\phi \mapsto \mathrm{free}_x(\phi)$$

by the following clauses

$$\mathrm{free}_x(p) := \begin{cases} 1 & , \text{if } x \in V(p) \\ 0 & , \text{ow,} \end{cases}$$

$$\mathrm{free}_x(\neg f) = \mathrm{free}_x(\phi)$$

$$\mathrm{free}_x(\phi \vee \psi) = \max\{\mathrm{free}_x(\phi), \mathrm{free}_x(\psi)\} = \mathrm{free}_x(\phi) \vee \mathrm{free}_x(\psi)$$

$$\mathrm{free}_x(\exists_{x_i}\phi) := \begin{cases} \mathrm{free}_x(\phi) & , \text{if } x \neq x_i \\ 0 & , x = x_i. \end{cases}$$

Finally we define "*x occurs free in ϕ*" if and only if $\mathrm{free}_x(\phi) = 1$ and "*ϕ is an L-sentence*", $\phi \in S(L)$, by

$$\begin{aligned} \phi \in S(L) &\leftrightarrow \forall_{x \in \mathrm{Var}_L}(x \text{ does not occur free in } \phi) \\ &\leftrightarrow \forall_{x \in \mathrm{Var}_L}(\mathrm{free}_x(\phi) = 0). \end{aligned}$$

Exercise 10. (i) Describe the form of an L-sentence with respect to the inductive definition of a formula.
(ii) Show that

(a) $\mathrm{free}_x(\phi \wedge \psi) = \mathrm{free}_x(\phi \vee \psi)$.
(b) $\mathrm{free}_x(\forall_{x_i}\phi) = \mathrm{free}_x(\exists_{x_i}\phi)$.

5. The definition of $\mathfrak{A} \models \phi_{\vec{x}}[a_1, \ldots, a_m]$

(I) The interpretation of terms. If $\mathfrak{A}$ is an L-structure, first one defines the interpretation $t_{\vec{x}}^{\mathfrak{A}}[a_1, \ldots, a_m] \in A$ of a term t whose variables are included in $\vec{x}$ and $a_1, \ldots, a_m \in A$, where the variables in $\vec{x}$ are pairwise different, while this is *not necessarily* the case for $a_1, \ldots, a_m$. Hence, the terms satisfy the condition

$$(\dagger) \quad V(t) \subseteq \{x_1, \ldots, x_m\}.$$

Note that there is no meaning to define this concept for a variable not included in $\vec{x}$. This is in contrast to the line of definition considered in [1], the main reference in the website of the lecture course. There one assigns an element of A to every variable first, while in the lecture course a more "local" approach is followed; we start from a term satisfying ($\dagger$) and **the general idea of the definition** is that each variable x_i in t is going to be interpreted by a_i. The definitional clauses are the following

$$x_i{}_{\vec{x}}^{\mathfrak{A}}[a_1, \ldots, a_m] := a_i$$

$$c_{\vec{x}}^{\mathfrak{A}}[a_1, \ldots, a_m] := c^{\mathfrak{A}}$$

$$(ft_1 \dots t_n)^{\mathfrak{A}}_{\vec{x}}[a_1, \dots, a_m] := f^{\mathfrak{A}} t_1{}^{\mathfrak{A}}_{\vec{x}}[a_1, \dots, a_m] \dots t_n{}^{\mathfrak{A}}_{\vec{x}}[a_1, \dots, a_m].$$

Note that the above is well-defined i.e.,

$$ft_1 \dots t_n \text{ satisfies } (\dagger) \to \forall_i (t_i \text{ satisfies } (\dagger)),$$

since

$$V(t) = \bigcup_{i=1}^{m} V(t_i) \subseteq \{x_1, \dots, x_m\} \to \forall_i (V(t_i) \subseteq \{x_1, \dots, x_m\}).$$

A trivial induction on terms shows that $t^{\mathfrak{A}}_{\vec{x}}[a_1, \dots, a_m] \in A$; the formula that we would prove by induction is

$$\forall_{t \in \mathrm{Term}_L} (V(t) \subseteq \{x_1, \dots, x_m\} \to t^{\mathfrak{A}}_{\vec{x}}[a_1, \dots, a_m] \in A).$$

If $m = 0$, then $V(t) = \emptyset$ and the above interpretation pertains only to terms with no variables. Hence, the interpretation $t^{\mathfrak{A}}_{\emptyset}$ which is equal to $t^{\mathfrak{A}}$ is exactly the following:

$$c^{\mathfrak{A}}_{\emptyset} := c^{\mathfrak{A}}$$

$$(ft_1 \dots t_n)^{\mathfrak{A}}_{\emptyset} := f^{\mathfrak{A}} t_1{}^{\mathfrak{A}}_{\emptyset} \dots t_n{}^{\mathfrak{A}}_{\emptyset}.$$

(II) The satisfaction relation. The next step is to extend the interpretation in $\mathfrak{A}$ from the terms of L to its formulas. Again we do that “locally”. Namely, we define the relation $\mathfrak{A} \models \phi_{\vec{x}}[a_1, \dots, a_m]$ for every formula ϕ satisfying

$$(*) \quad \mathrm{free}_x(\phi) = 1 \to x \in \{x_1, \dots, x_m\}.$$

The writing of the definitional clauses stresses the difference between the syntactical $=, \neg, \vee, \exists$ of the language L and the semantical “equality”, “not”, “or”, “there exists” of our metalanguage (e.g., this could be the metalanguage of set theory).

(G1) $\mathfrak{A} \models (t_1 = t_2)_{\vec{x}}[a_1, \dots, a_m]$ iff $t_1{}^{\mathfrak{A}}_{\vec{x}}[a_1, \dots, a_m] = t_2{}^{\mathfrak{A}}_{\vec{x}}[a_1, \dots, a_m]$.

By the previous analysis we have that $t_1{}^{\mathfrak{A}}_{\vec{x}}[a_1, \dots, a_m], t_2{}^{\mathfrak{A}}_{\vec{x}}[a_1, \dots, a_m] \in A$ and the definition says that these two elements of A are required to be equal. Note that

$$\mathrm{free}_x(t_1 = t_2) = 1 \leftrightarrow x \in V(t_1) \cup V(t_2) \subseteq \{x_1, \dots, x_m\}$$

shows that the above definition is well-defined.

(G2) $\mathfrak{A} \models (Rt_1 \dots t_n)_{\vec{x}}[a_1, \dots, a_m]$ iff $R^{\mathfrak{A}}(t_1{}^{\mathfrak{A}}_{\vec{x}}[a_1, \dots, a_m] \dots t_n{}^{\mathfrak{A}}_{\vec{x}}[a_1, \dots, a_m])$

i.e., the n-tuple $(t_1{}^{\mathfrak{A}}_{\vec{x}}[a_1, \dots, a_m] \dots t_1{}^{\mathfrak{A}}_{\vec{x}}[a_1, \dots, a_m]) \in R^{\mathfrak{A}}$. Again the inclusion

$$V(Rt_1, \dots, t_n) = \bigcup_{i=1}^{n} V(t_i) \subseteq \{x_1, \dots, x_m\}$$

shows that (G2) is well-defined.

(G3) $\mathfrak{A} \models \neg\phi_{\vec{x}}[a_1, \ldots, a_m]$ iff **not** $\mathfrak{A} \models \phi_{\vec{x}}[a_1, \ldots, a_m]$.

Since

$$\text{free}_x(\neg\phi) = 1 \leftrightarrow \text{free}_x(\phi) = 1,$$

(G3) is well-defined i.e., if $\neg\phi$ satisfies $(*)$, then ϕ satisfies $(*)$.

(G4) $\mathfrak{A} \models (\phi \vee \psi)_{\vec{x}}[a_1, \ldots, a_m]$ iff $\mathfrak{A} \models \phi_{\vec{x}}[a_1, \ldots, a_m]$ **or** $\mathfrak{A} \models \psi_{\vec{x}}[a_1, \ldots, a_m]$.

Since $\text{free}_x(\phi \vee \psi) = \text{free}_x(\phi) \vee \text{free}_x(\psi)$, then $\text{free}_x(\phi) = 1 \to \text{free}_x(\phi \vee \psi) = 1$, which implies that $x \in \{x_1, \ldots, x_m\}$, by our hypothesis on $(\phi \vee \psi)$. We work similarly with ψ and we conclude that (G4) is also well-defined i.e.,

$$(\phi \vee \psi) \text{ satisfies } (*) \to \phi, \psi \text{ satisfy } (*).$$

(G5a) $\mathfrak{A} \overset{z \in \vec{x}}{\models} (\exists_z \psi)_{\vec{x}}[a_1, \ldots, a_m]$ iff **there exists some** $b \in A$ such that

$$\mathfrak{A} \models \psi_{\vec{x}}[a_1, \ldots, a_{i-1}, b, a_{i+1}, \ldots, a_m].$$

We show that in case $z = x_i$, for some i, if $\exists_z \psi$ satisfies the $(*)$ condition, so does ψ; if $\text{free}_{x_i}(\psi) = 1$, then we get automatically the required conclusion. If $x \neq x_i$ and $\text{free}_x(\psi) = 1$, then we get that $\text{free}_x(\exists_z \psi) = 1$, therefore by our hypothesis on $\exists_z \psi$ we get that $x \in \{x_1, \ldots, x_m\}$.
The general idea of the definition is that in the interpretation-satisfaction of the formula ϕ the object a_i takes the place of every free occurrence of x_i in ϕ. If $x_i = z$ does not occur free, as in the case of $\exists_{x_i} \psi$, we require that some object b takes the place of x_i in ψ. Of course, the clause (G5a) shows that although $\vec{x}$ remains the same, we are reduced to the relation $\mathfrak{A} \models \psi_{\vec{x}}[a_1, \ldots, a_{i-1}, b, a_{i+1}, \ldots, a_m]$ w.r.t. the possibly new element $(a_1, \ldots, a_{i-1}, b, a_{i+1}, \ldots, a_m)$ of A^m. But then we use the same pattern-definition. This is a kind of "nestedness" of the general definition.

(G5b) $\mathfrak{A} \overset{z \notin \vec{x}}{\models} (\exists_z \psi)_{\vec{x}}[a_1, \ldots, a_m]$ iff **there exists some** $b \in A$ such that

$$\mathfrak{A} \models \psi_{z, \vec{x}}[b, a_1, \ldots, a_m].$$

The only variable which can appear free in ψ and not in $\exists_z \psi$ is z. Since by hypothesis $z \notin \vec{x}$, we need to expand the vector of variables $\vec{x}$ to $z, \vec{x}$, in order to have a meaningful definition i.e.,

$$(*) \to (**)$$

$$(**) \quad \text{free}_x(\psi) = 1 \to x \in \{z, x_1, \ldots, x_m\}.$$

In accordance to the general idea of the definition, since z cannot be interpreted by some a_i, since each a_i replaces the free occurrence of x_i and $z \notin \vec{x}$, we need a new object in A to replace z. Since a formula is a finite object, the above procedure ends i.e., although we are reduced in this case to larger vectors, this cannot go on for ever because of the "finiteness" of the formulas. This is easier to see if you work with specific examples of formulas.

Next we give two special cases of the above definition:

(A) $m = 1$. The corresponding $(*)$-condition and clauses are:

$$(*) \quad \mathrm{free}_y(\phi) = 1 \to y = x.$$

(G1) $\mathfrak{A} \models (t_1 = t_2)_x[a]$ iff ${t_1}_x^{\mathfrak{A}}[a] = {t_2}_x^{\mathfrak{A}}[a]$.

(G2) $\mathfrak{A} \models (Rt_1 \ldots t_n)_x[a]$ iff $R^{\mathfrak{A}}({t_1}_x^{\mathfrak{A}}[a] \ldots {t_n}_x^{\mathfrak{A}}[a])$.

(G3) $\mathfrak{A} \models \neg\phi_x[a]$ iff **not** $\mathfrak{A} \models \phi_x[a]$.

(G4) $\mathfrak{A} \models (\phi \vee \psi)_x[a]$ iff $\mathfrak{A} \models \phi_x[a]$ **or** $\mathfrak{A} \models \psi_x[a]$.

(G5a) $\mathfrak{A} \overset{z=x}{\models} (\exists_z \psi)_x[a]$ iff **there exists some** $b \in A : \mathfrak{A} \models \psi_x[b]$.

(G5b) $\mathfrak{A} \overset{z\neq x}{\models} (\exists_z \psi)_x[a]$ iff **there exists some** $b \in A : \mathfrak{A} \models \psi_{z,x}[b, a]$.

(B) $m = 0$. The corresponding $(*)$-condition is that $\phi \in S(L)$ i.e., ϕ is a **sentence**. The definitional clauses are:

(G1) $\mathfrak{A} \models t_1 = t_2$ iff ${t_1}^{\mathfrak{A}} = {t_2}^{\mathfrak{A}}$.

(G2) $\mathfrak{A} \models Rt_1 \ldots t_n$ iff $R^{\mathfrak{A}}({t_1}^{\mathfrak{A}} \ldots {t_n}^{\mathfrak{A}})$.

(G3) $\mathfrak{A} \models \neg\phi$ iff **not** $\mathfrak{A} \models \phi$.

(G4) $\mathfrak{A} \models (\phi \vee \psi)$ iff $\mathfrak{A} \models \phi$ **or** $\mathfrak{A} \models \psi$.

(G5s) $\mathfrak{A} \models (\exists_x \psi)$ iff **there exists some** $b \in A : \mathfrak{A} \models \psi_x[b]$.

Note that when $m = 0$, both clauses (G5a) and (G5b) are reduced to (G5s), and that since every free occurrence of x in ϕ has been replaced by b, no "capture" is possible in the direction "if" of the last clause.

Exercise 11. If ϕ is a sentence of L and $\mathfrak{A}$ is an L-structure, check the validity or not of the following propositions:

(i) not ($\mathfrak{A} \models \phi$ and $\mathfrak{A} \models \neg\phi$).
(ii) ($\mathfrak{A} \models \phi$) or ($\mathfrak{A} \models \neg\phi$).

Exercise 12. If $\mathfrak{A}$ is an L-structure and $a \in A$, show the following:

(i) If ϕ, ψ are sentences of L, then

$$\mathfrak{A} \models \phi \to \psi \text{ iff } \textbf{if } \mathfrak{A} \models \phi, \textbf{ then } \mathfrak{A} \models \psi.$$

(ii) Does a similar condition hold for $\mathfrak{A} \models (\phi \to \psi)_x[a]$?
(iii) $\mathfrak{A} \models (\forall_x \phi \to \phi)_x[a]$.
(iv) $\mathfrak{A} \models (\forall_x \phi \to \exists_x \phi)_x[a]$.

★ **Exercise 13**. An L-formula is called *positive*, if it contains no $\neg$ symbol.

Show that for every positive L-formula there is an L-structure which satisfies it.

6. L-expansions and the relation of logical consequence

Exercise 14. Let $\overline{L} = \{R\}$, where R is a two-place relation symbol, and $\overline{\mathfrak{A}} = (\mathbb{N}, R^{\overline{\mathfrak{A}}})$ is an $\overline{L}$-Structure, where $\mathbb{N} = \{0, 1, 2, \dots\}$ and

$$R^{\overline{\mathfrak{A}}} = \{(m, n) \mid m, n \in \mathbb{N} \text{ and } m < n\}.$$

If f is an one-place function symbol and $L = \overline{L} \cup \{f\}$, then

(a) Find an L-expansion of $\overline{\mathfrak{A}}$ such that the following L-sentence holds

$$\forall x \exists y \exists z (R(x, y) \wedge R(y, z) \wedge R(f(z), f(y))).$$

(b) In every L-expansion of $\overline{\mathfrak{A}}$ the following L-sentence holds

$$\forall x \exists y \exists z (R(x, y) \wedge R(y, z) \wedge \neg R(f(z), f(y))).$$

If $T \subseteq S(L)$ and $\phi \in S(L)$, where $S(L)$ denotes the sentences of L, then we define the relation $T \models \phi$, "ϕ is a logical consequence of T", by

$$T \models \phi :\leftrightarrow \forall_{\mathfrak{A}} (\mathfrak{A} \models T \rightarrow \mathfrak{A} \models \phi).$$

Similarly we define $\Phi \models \phi$, where $\Phi \subseteq \mathrm{Form}_L$ and $\phi \in \mathrm{Form}_L$.

Exercise 15. If ϕ is a sentence of L, $\mathfrak{A}$ is an L-structure, and T is an L-theory, check the validity or not of the following:

$$(T \models \phi) \vee (T \models \neg\phi).$$

Exercise 16. Check if for all L-sentences ϕ, ψ and θ the following hold or not:
(a) $(\phi \vee \psi) \models \theta$ if and only if $(\phi \models \theta$ and $\psi \models \theta)$.
(b) $(\phi \wedge \psi) \models \theta$ if and only if $(\phi \models \theta$ or $\psi \models \theta)$.

Exercise 17. Show that

$$\exists_x \forall_y \phi \models \forall_y \exists_x \phi.$$

Find an example of such a logical consequence from standard mathematics. What about the converse logical consequence?

7. The relation of logical equivalence, prenex and Skolem normal forms

If $\phi, \psi \in \text{Form}$, we define $\phi \models\mid \psi$, "ϕ is logically equivalent to ψ", by

$$\begin{aligned} \phi \models\mid \psi &\leftrightarrow \phi \models \psi \text{ and } \psi \models \phi \\ &\leftrightarrow \forall_{\mathfrak{A}}(\mathfrak{A} \models \phi \text{ if and only if } \mathfrak{A} \models \psi). \end{aligned}$$

Exercise 18. Show the following:

(i) $\forall_x \forall_y \phi \models\mid \forall_y \forall_x \phi$.
(ii) $\exists_x \exists_y \phi \models\mid \exists_y \exists_x \phi$.
(iii) $\forall_x \phi \models\mid \phi$, if $x \notin \text{FV}(\phi)$.
(iv) $\exists_x \phi \models\mid \phi$, if $x \notin \text{FV}(\phi)$.
(v) $\forall_x (\phi \wedge \psi) \models\mid \forall_x \phi \wedge \forall_x \psi$.
(vi) $\exists_x (\phi \vee \psi) \models\mid \exists_x \phi \vee \exists_x \psi$.
(vii) $\forall_x (\phi \vee \psi) \models\mid \forall_x \phi \vee \psi$, if $x \notin \text{FV}(\psi)$.
(viii) $\exists_x (\phi \wedge \psi) \models\mid \exists_x \phi \wedge \psi$, if $x \notin \text{FV}(\psi)$.

Exercise 23. Show the following:

(i) If $\phi \models\mid \psi$, then $\neg\phi \models\mid \neg\psi$.
(ii) If $\phi \models\mid \psi$ and $\phi' \models\mid \psi'$, then $\phi \vee \phi' \models\mid \psi \vee \psi'$.
(iii) If $\phi \models\mid \psi$, then $\exists_x \phi \models\mid \exists_x \psi$.

We define the set $\text{Sub}(\phi)$ of all subformulas of some formula ϕ recursively by:

$$\text{Sub}(p) := \{p\},$$

$$\text{Sub}(\neg\phi) := \text{Sub}(\phi) \cup \{\neg\phi\},$$
$$\text{Sub}(\phi \vee \psi) := \text{Sub}(\phi) \cup \text{Sub}(\psi) \cup \{\phi \vee \psi\},$$
$$\text{Sub}(\exists_x \phi) := \text{Sub}(\phi) \cup \{\exists_x \phi\},$$

where $p \in \text{Prim}$, the set of prime formulas in Form.

Exercise 19. If $\sigma \in \text{Sub}(\phi)$ and $\sigma' \in \text{Form}$, then

$$\text{if } \sigma \models\mid \sigma', \text{ then } \phi[\sigma'/\sigma] \models\mid \phi,$$

where $\phi[\sigma'/\sigma]$ is the formula resulting by substitution in ϕ of σ by σ'.

Exercise 20. Find the prenex normal form of the following formulas:

(i) $\forall_x (Rx \to \forall_y S(x, y))$.

(ii) $\forall_x(Rx \to \exists_y S(x,y))$.
(iii) $\neg\exists_x Rx \vee \forall_x Sx$.

Exercise 21. Find formulas ϕ, ψ which show the necessity of the variable condition in Exercise 22(iii) (iv) (vii), and (viii), respectively.

Exercise 22. **(a)** In the exercise 3 of Blatt 4 we saw that a satisfiable pure $\forall$-sentence without function symbols is also satisfiable by a finite structure. Find a satisfiable pure $\forall$-sentence with two function symbols which cannot be satisfied by a finite structure.
(b) Is there a satisfiable pure $\forall$-sentence with one function symbols which cannot be satisfied by a finite structure?

8. The logical closure operator

If X is a set, and $\mathcal{P}(X)$ denotes the power set of X, a *closure operator* on X is a function C

$$C : \mathcal{P}(X) \to \mathcal{P}(X)$$
$$A \mapsto C(A),$$

satisfying the following properties:

(i) $A \subseteq C(A)$.
(ii) $A_1 \subseteq A_2 \to C(A_1) \subseteq C(A_2)$.
(iii) $C(C(A)) = C(A)$.

Exercise 23. Show that a closure operator C on X satisfies the following properties (actually the only necessary condition is (ii)):

(i) $\bigcup_{i\in I} C(A_i) \subseteq C(\bigcup_{i\in I} A_i)$.
(ii) $C(\bigcap_{i\in I} A_i) \subseteq \bigcap_{i\in I} C(A_i)$,

where $(A_i)_{i\in I}$ is a family of subsets of X indexed by a set I.

If we fix a 1st-order language L, then we define the *logical* operator C on $S(L)$ (with respect to our fixed language L) as the function

$$C : \mathcal{P}(S(L)) \to \mathcal{P}(S(L))$$
$$T \mapsto C(T),$$

and

$$C(T) := \{\phi \in S(L) \mid T \models \phi\}.$$

Exercise 24 . Show that the logical operator C is a closure operator on $S(L)$.

A theory T is called *closed*, if $C(T) = T$, which is equivalent, because of property (i) of a closure operator, to $C(T) \subseteq T$, in other words,

$$T \models \phi \to \phi \in T.$$

A special closure operator which is fundamental in general topology is defined as follows: If X is a set, a *topological closure operator* on X is a function Cl

$$\mathrm{Cl} : \mathcal{P}(X) \to \mathcal{P}(X)$$

$$A \mapsto \mathrm{Cl}(A),$$

satisfying the following properties:

(i) Cl is a closure operator.
(ii) $\mathrm{Cl}(\emptyset) = \emptyset$.
(iii) $\mathrm{Cl}(A \cup B) = \mathrm{Cl}(A) \cup \mathrm{Cl}(B)$.

It is direct to see that the set

$$\mathcal{T}_{\mathrm{Cl}} := \{X \setminus \mathrm{Cl}(A) \mid A \subseteq X\}$$

is a topology on X, and that the sets of the form $A = \mathrm{Cl}(A)$ are the *closed* sets with respect to this topology. Conversely, if $(X, \mathcal{T})$ is a topological space, then the operator $\mathrm{Cl}_{\mathcal{T}}$ defined by

$$A \mapsto \overline{A},$$

where $A \subseteq X$ and $\overline{A}$ denotes the $\mathcal{T}$-closure of A i.e., the least $\mathcal{T}$-closed set including A, is a topological closure operator on X. We call the operator $\mathrm{Cl}_{\mathcal{T}}$ the topological closure operator *induced* by the topology $\mathcal{T}$.

Exercise 25. Find a (simple) topological space $(X, \mathcal{T})$ the induced topological closure operator $\mathrm{Cl}_{\mathcal{T}}$ of which does not satisfy the equalities in cases (i) and (ii) of Exercise 22, respectively.

Thus, although a topological closure operator satisfies more properties than the monotonicity condition (ii) of its definition, still these equalities do not hold in general.

9. Kinds of 1st-order theories

A theory T is called *consistent*, if

$$\forall_{\phi \in S(L)}(T \not\models (\phi \wedge \neg\phi)) \leftrightarrow \nexists_{\phi \in S(L)}(T \models \phi \wedge \neg\phi).$$

A theory T is called *inconsistent*, if

$$\exists_{\phi \in S(L)}(T \models (\phi \wedge \neg\phi)).$$

A theory T is called *complete*, if

$$\forall_{\phi \in S(L)}(T \models \phi \vee T \models \neg\phi).$$

Obviously, a closed theory is complete, if

$$\forall_{\phi \in S(L)}(\phi \in T \vee \neg\phi \in T).$$

A theory T is called *incomplete*, if

$$\exists_{\phi \in S(L)}(T \not\models \phi \wedge T \not\models \neg\phi).$$

A theory T is called *finitely axiomatizable*, if

$$\exists_{F \subseteq^{\mathrm{fin}} T}(T = C(F)),$$

where $F \subseteq^{\mathrm{fin}} T$ denotes that F is a finite subset of T.

Exercise 26. By the definition of the logical closure operator we have that

$$C(\emptyset) = \{\phi \in S(L) \mid \emptyset \models \phi\} = \{\text{valid sentences}\}.$$

(i) If T is a 1st-order theory, then

$$C(\emptyset) \subseteq C(T) \subseteq S(L),$$

and C does not satisfy condition (iv) of a topological closure operator.
(ii) Find theories T_1, T_2 satisfying

$$C(T_1) \cup C(T_2) \subsetneq C(T_1 \cup T_2),$$

i.e., C does not satisfy condition (v) of a topological closure operator.
(iii) The theory $S(L)$ is the maximum closed L-theory, and it is also inconsistent.
(iv) The theory $S(L)$ is the only inconsistent closed L-theory.
(v) Give an example of an incomplete theory.

10. Some basic 1st-order theories

For simplicity we do not write the following axioms with the corresponding universal quantifiers at their beginning.

1. **Peano Arithmetic** PA.
Language: $L = (+, \cdot, S, 0)$.
Axioms:

1. $\neg S(x) = 0$.
2. $S(x) = S(y) \to x = y$.
3. $x + 0 = x$.
4. $x + S(y) = S(x + y)$.
5. $x \cdot 0 = 0$.
6. $x \cdot S(y) = x \cdot y + x$.

7_ϕ. $\phi(0) \to \forall_x(\phi(x) \to \phi(S(x))) \to \forall_x \phi(x)$.

2. **Partial Order** O.
Language: $L = (<)$.

Axioms:

1. $\neg x < x$.
2. $x < y \to y < z \to x < z$.
3. $x < y \to \neg y < x$.

3. **Linear Order** LO.
Language: $L = (<)$.

Axioms:

1. O.
2. $x < y \vee x = y \vee y < x$.

4. **Dense Linear Order** DLO.
Language: $L = (<)$.

Axioms:

1. LO.
2. $x < y \to \exists_z(x < z \wedge z < y)$.

5. **Dense Linear Order without Endpoints** DLO*.
Language: $L = (<)$.

Axioms:

1. DLO.
2. $\forall_x \exists_{y,z}(y < x \wedge x < z)$.

6. **Ring** R.
Language: $L = (+, \cdot, 0)$.

Axioms:

1. Abelian group w.r.t. $+$.
2. $x \cdot (y \cdot z) = (x \cdot y) \cdot z$.
3. $x \cdot (y + z) = x \cdot y + x \cdot z$.
4. $(x + y) \cdot z = x \cdot z + y \cdot z$.

7. **Commutative Rings with Unit** CR.
Language: $L = (+, \cdot, 0, 1)$.

Axioms:

1. R.
2. $x \cdot y = y \cdot x$.
3. $1 \cdot x = x$.

8. **Integral Domain** ID.
Language: $L = (+, \cdot, 0, 1)$.

Axioms:

1. CR.
2. $x \cdot y = 0 \rightarrow x = 0 \vee y = 0$.

9. **Field** F.
Language: $L = (+, \cdot, 0, 1)$.

Axioms:

1. ID.
2. $x \neq 0 \rightarrow \exists_y (x \cdot y = 1)$.

10. **Field of characteristic** p F(p).
Language: $L = (+, \cdot, 0, 1)$.

Axioms:

1. F.
2. $p \cdot 1 = 0 \wedge (p-1) \cdot 1 \neq 0 \wedge \ldots \wedge 2 \cdot 1 \neq 0$.

11. **Field of characteristic** 0 F(0).
Language: $L = (+, \cdot, 0, 1)$.

Axioms:

1. F.

2_p. $p \cdot 1 \neq 0$.

12. **Ordered Field** OF.
Language: $L = (<, +, \cdot, 0, 1)$.

Axioms:

1. F.
2. LO.
3. $x < y \rightarrow x + z < y + z$.
4. $0 < x \rightarrow 0 < y \rightarrow 0 < x \cdot y$.

13. **Real closed Field** RCF.
Language: $L = (<, +, \cdot, 0, 1)$.

Axioms:

1. OF.
2. $0 < x \rightarrow \exists_y (x = y \cdot y)$.

3_{2n+1}. $x_{2n+1} \neq 0 \rightarrow \exists_y (x_{2n+1} y^{2n+1} + x_{2n} y^{2n} + \ldots + x_1 y + x_0 = 0)$.

14. **Algerbraic Closed Field** ACF.
Language: $L = (+, \cdot, 0, 1)$.

Axioms:
1. F.
2_n. $x_n \neq 0 \to \exists_y (x_n y^n + x_{n-1} y^{n-1} + \ldots + x_1 y + x_0 = 0)$.

11. The finite spectrum of a sentence (Aufgabe 1, Blatt 5)

If ϕ is an Ł-sentence, then its *finite spectrum* is defined by

$$S(\phi) := \{|A| \mid \mathcal{A} \models \phi, A \text{ is finite}, \mathcal{A} \text{ is an } L\text{-structure}\}.$$

If $X \subseteq \mathbb{N}$, it is called a *spectrum*, if there is some 1st-order language L and some L-sentence ϕ such that

$$X = S(\phi).$$

Exercise 27. Show that the following subsets of $\mathbb{N}$ are spectra.

(a) $X_1 = \{2n + 1 \mid n \in \mathbb{N}\}$.
(b) $X_2 = \{an + b \mid n \in \mathbb{N}\}$, where $a, b \in \mathbb{N}$.
(c) $X_3 = \{n^2 \mid n \in \mathbb{N}\}$.
(d) $X_4 = \{n^2 + 1 \mid n \in \mathbb{N}\}$.
(e) $X_5 = \{n \in \mathbb{N} \mid n \text{ is a composite number}\}$.
(f) $X_6 = \{p^n \mid n \in \mathbb{N}, p \text{ is a prime number}\}$.

Exercise 28. Show that the intersection or union of two spectra is a spectrum. (it is an open problem if the complement of a spectrum is a spectrum).

12. Complete theories

Exercise 29. Give an example of an incomplete theory and an example of a complete theory.

If $\mathfrak{A}, \mathfrak{B}$ are L-structures we say that they are are *elementarily equivalent*, $\mathfrak{A} \equiv \mathfrak{B}$, if they satisfy the same L-sentences. Actually the following hold:

$$\begin{aligned}
\mathfrak{A} \equiv \mathfrak{B} &\leftrightarrow \mathrm{Th}(\mathfrak{A}) = \mathrm{Th}(\mathfrak{B}) \\
&\leftrightarrow \forall_{\phi \in S(L)} (\mathfrak{A} \models \phi \leftrightarrow \mathfrak{B} \models \phi) \\
&\leftrightarrow \mathfrak{A} \models \mathrm{Th}(\mathfrak{B}) \\
&\leftrightarrow \mathfrak{B} \models \mathrm{Th}(\mathfrak{A}).
\end{aligned}$$

Exercise 30 Show that $\mathfrak{A} \cong \mathfrak{B} \to \mathfrak{A} \equiv \mathfrak{B}$, and give a counterexample to the converse implication (**Hint**: use exercise 4, Blatt 7).

★ **Exercise 31**. Suppose that $(A, <)$ and $(B, <)$ are are two dense linear orders without endpoints (we keep for simplicity the same symbol for the order). Then the following hold:

(i) There cannot be a finite dense linear order without endpoints i.e., the carrier set cannot be finite.
(ii) Give an example of a countable dense linear order without endpoints.
(iii) Give an example of an uncountable dense linear order without endpoints.
(iv) If A, B are countable, then there exists $e : A \to B$ which is 1-1 and onto B satisfying

$$\forall_{a_1, a_2 \in A}(a_1 < a_2 \leftrightarrow e(a_1) < e(a_2)).$$

(v) If A, B are countable $a \in A$ and $b \in B$, then there exists $e_{ab} : A \to B$ which is 1-1 and onto B satisfying

$$\forall_{a_1, a_2 \in A}(a_1 < a_2 \leftrightarrow e_{ab}(a_1) < e_{ab}(a_2)),$$

and

$$e_{ab}(a) = b.$$

13. Sequent Calculus

Exercise 32. Which of the following rules is correct? If so, find a derivation of it.

$$\frac{\Gamma\phi_1\psi_1 \quad \Gamma\phi_2\psi_2}{\Gamma(\phi_1 \vee \phi_2)(\psi_1 \vee \psi_2)},$$

$$\frac{\Gamma\phi_1\psi_1 \quad \Gamma\phi_2\psi_2}{\Gamma(\phi_1 \vee \phi_2)(\psi_1 \wedge \psi_2)}.$$

Exercise 33. Show that the following rules are derivable:

$$\frac{}{\phi \vee \neg\phi},$$

$$\frac{\Gamma\phi\psi}{\Gamma\neg\psi\neg\phi},$$

$$\frac{\Gamma(\phi \vee \psi) \quad \Gamma\neg\phi}{\Gamma\psi},$$

$$\frac{\Gamma(\phi \to \psi) \;\; \Gamma\phi}{\Gamma\psi},$$

$$\frac{\Gamma\phi}{\Gamma\neg\neg\phi},$$

$$\frac{\Gamma\phi}{\Gamma\exists_x\phi}.$$

$$\frac{\Gamma\phi \; \psi}{\Gamma\exists_x\phi \; \psi}, \quad x \text{ not free in } \Gamma\psi.$$

$$\frac{\Gamma\phi}{\Gamma\forall_x\phi}, \quad x \text{ not free in } \Gamma.$$

Exercise 34. Derive the following sequents:

$$\phi \; (\phi \vee \psi),$$

$$(\phi \vee \psi) \; \neg\phi \; \psi.$$

Exercise 35. If T is an L-theory we define

$$T^* = \{\phi \in S(L) \mid T \vdash_L \phi\}.$$

If $\phi \in S(L)$ show that

$$T^* \vdash_L \phi \to T \vdash_L \phi.$$

14. Recursive functions

As in the lecture course notes we assume the principle of the excluded middle, and the characteristic function K_A of some $A \subseteq \mathbb{N}^k$, where $k \geq 1$, is defined by

$$K_A(\vec{n}) := \begin{cases} 0 & \text{, if } A(\vec{n}) \\ 1 & \text{, ow.} \end{cases}$$

Also, the projection functions $I_i^k : \mathbb{N}^k \to \mathbb{N}$ are defined by

$$I_i^k(\vec{n}) = I_i^k(n_1, \ldots, n_k) = n_i,$$

for every $i \in \{1, \ldots, k\}$. Note that $I_1^1 = \mathrm{id}_{\mathbb{N}}$.

A *recursive* function $f : \mathbb{N}^2 \to \mathbb{N}$ and a *recursive* function $f : \mathbb{N}^k \to \mathbb{N}$, or

better the sets Rek^2 and Rek^k, where $k \geq 1$, are defined *simultaneously* by the following inductive rules:
(I)

$$\frac{}{+ \in \mathrm{Rek}^2}, \quad \frac{}{\cdot \in \mathrm{Rek}^2}, \quad \frac{}{K_< \in \mathrm{Rek}^2},$$

$$\frac{}{I_i^k \in \mathrm{Rek}^k}, \quad 1 \leq i \leq k$$

(II)

$$\frac{G \in \mathrm{Rek}^n, \quad H_1, \ldots, H_n \in \mathrm{Rek}^k}{\mathrm{Comp}(G, H_1, \ldots, H_n) \in \mathrm{Rek}^k},$$

where

$$\mathrm{Comp}(G, H_1, \ldots, H_n)(n_1, \ldots, n_k) = G(H_1(n_1, \ldots, n_k), \ldots, H_n(n_1, \ldots, n_k)).$$

(III)

$$\frac{G \in \mathrm{Rek}^{k+1}_{\mathrm{tot0}}}{\mu G \in \mathrm{Rek}^k}, \quad n \geq 1,$$

where

$$\mathrm{Rek}^{k+1}_{\mathrm{tot0}} = \{f \in \mathrm{Rek}^{k+1} \mid \forall_{\vec{n} \in \mathbb{N}^k} \exists_m (f(\vec{n}, m) = 0\}$$

and

$$\mu G(\vec{n}) := \mu m.G(\vec{n}, m) = 0.$$

Note 1: If we combine (I) and (II) we get e.g., that $H_1 + H_2, H_1 \cdot H_2 \in \mathrm{Rek}^k$.
Note 2: Every recursive function $f \in \mathrm{Rek}^k$ defined above is *total* i.e., its domain is the whole set $\mathbb{N}^k$. If we drop the totality condition in (III), we get the *partial* recursive functions.
Note 3: Exactly because the recursive functions are defined inductively, if we want to show that for every $f \in \mathrm{Rek}^k$ we have $P(f)$, where P is any formula on functions, we use the induction axiom that corresponds to the above definition (Exercise 3, Blatt 9).
Note 4: Verify that every rule of the main definition expresses an **algorithm** for finding the output $f(\vec{n})$, given the input $\vec{n}$. The non-trivial fact is that every algorithmic function (this is an intuitive notion) is recursive (**Church-Turing Thesis**)!!!
Note 5: Each set Rek^k is countable (but there is no recursive enumeration of it). Thus, most of the functions $\mathbb{N}^k \to \mathbb{N}$ are *not* recursive.

You can use the following facts without proof in the exercises and in the Klausur:

(I) If $A \subseteq \mathbb{N}^n$ is recursive and $H_1, \ldots, H_n \in \mathrm{Rek}^k$, then

$$B = \mathrm{Comp}(A, H_1, \ldots, H_n) \text{ recursive } \subseteq \mathbb{N}^k,$$

where

$$B(\vec{n}) \leftrightarrow A(H_1(\vec{n}), \ldots, H_n(\vec{n})).$$

(II) If A is a total recursive relation i.e., $\forall_{\vec{n}} \exists_m A(\vec{n}, m)$, then

$$F(\vec{n}) := \mu m.A(\vec{n}, m)$$

is recursive.

(III) Recursiveness is closed under arbitrary compositions.

(IV) The constant functions $F_m : \mathbb{N}^k \to \mathbb{N}$, $\vec{n} \mapsto m$, are recursive.

(V) Recursiveness is closed under complements, $\cap$ and $\cup$.

(VI) $\geq, \leq, =, >, <$ are recursive.

(VII) Bounded quantification preserves recursiveness i.e., if A is recursive, then

$$B(\vec{n}, m) \leftrightarrow \exists_{k<m} A(\vec{n}, k)$$

$$C(\vec{n}, m) \leftrightarrow \forall_{k<m} A(\vec{n}, k)$$

are recursive.

(VIII) The definition by cases preserves recursiveness e.g., if $R \subseteq \mathbb{N}^k$ is recursive, and $G_1, G_2 \in \mathrm{Rek}^k$, then

$$F(\vec{n}) := \begin{cases} G_1(\vec{n}) & \text{, if } R(\vec{n}) \\ G_2(\vec{n}) & \text{, ow} \end{cases}$$

is in Rek^k.

(IX) The functions $\pi : \mathbb{N}^2 \to \mathbb{N}$ and $\pi_1, \pi_2 : \mathbb{N} \to \mathbb{N}$ are recursive.

(X) The modified subtraction $\dot{-}$ and the remainder $r(m, n)$ are recursive.

(XI) The Wertverlaufsatz and especially the *primitive recursion rule*

$$\frac{b_1 \in \mathrm{Rek}^k, \quad b_2 \in \mathrm{Rek}^{k+2}}{f = \mathrm{Rek}(b_1, b_2) \in \mathrm{Rek}^{k+1}},$$

where

$$f(0, \vec{n}) = b_1(\vec{n})$$

$$f(m+1, \vec{n}) = b_2(f(m, \vec{n}), m, \vec{n}).$$

If we consider no parameters, we get the following:

$$f(0) = b_1 \in \mathbb{N}$$

$$f(m+1) = b_2(f(m), m).$$

Exercise 36. Show that the following functions are recursive:

(i) The predecessor of n.
(ii) $f(n, m) = |n - m|$.

(iii) $g(n,m) = \min(n,m)$.
(iv) $h(n,m) = \max(n,m)$.
(v) The factorial function $f(n) = n!$.
(vi) The quotient $q(x,y)$ of the division of y by x, where $x \neq 0$.

Exercise 37. (i) Using the primitive recursion rule show that if $f : \mathbb{N}^{n+1} \to \mathbb{N}$ is recursive, then the functions

$$g(m, \vec{a}) = \sum_{i \leq m} f(i, \vec{a})$$

$$h(m, \vec{a}) = \prod_{i \leq m} f(i, \vec{a})$$

are recursive.
(ii) If $G : \mathbb{N}^{n+1} \to \mathbb{N}$ is a recursive function, then the function $F : \mathbb{N}^{n+1} \to \mathbb{N}$ defined by

$$F(\vec{a}, m) = \begin{cases} \mu n \leq m . G(\vec{a}, n) = 0 & \text{, if } \exists_{n \leq m}(G(\vec{a}, n) = 0) \\ 0 & \text{, ow} \end{cases}$$

is recursive (do not use the μ-operator scheme).

Exercise 38. Let $f(n) = g_1(n)$, if n is a perfect cube, and $f(n) = g_2(n)$ otherwise. Show that if g_1, g_2 are recursive, then f is also recursive.

Exercise 39. Let

$$h_1(0,n) = f_1(n)$$

$$h_2(0,n) = f_2(n)$$

$$h_1(m+1,n) = g_1(h_1(m,n), h_2(m,n), n)$$

$$h_2(m+1,n) = g_2(h_1(m,n), h_2(m,n), n)$$

Show that if f_1, f_2, g_1, g_2 are recursive, then h_1, h_2 are recursive.

Exercise 40. Show that the function of the Fibonacci numbers

$$f(0) = 0$$

$$f(1) = 1$$

$$f(n+2) = f(n) + f(n+1)$$

is recursive.

15. Recursively enumerable sets

An $R \subseteq \mathbb{N}^n$ is called *recursively enumerable* (r.e.) iff

$$\exists_{Q \subseteq \mathbb{N}^{n+1}}(Q \in \mathrm{Rek}^{n+1} \wedge \forall_{\vec{a}}(R(\vec{a}) \leftrightarrow \exists_{b \in \mathbb{N}}(Q(\vec{a}, b)))).$$

Verify that from the above definition we have an algorithm for answering only Yes in the question if $R(\vec{a})$, and not generally one for answering No.

Exercise 41. Show the following:

(i) $\mathbb{N}$ is recursive.
(ii) If A, B are recursive sets, then $A \times B$ is recursive.
(iii) If R is recursive, then $R, \mathbb{N}^n \setminus R$ are recursively enumerable.
(iv) If A, B are recursively enumerable sets, then $A \times B$ is recursively enumerable.

Exercise 42. (i) Show that if $K \subseteq \mathbb{N}$ is recursively enumerable and not recursive, then K and $\mathbb{N} \setminus K$ are infinite.
(ii) If $Q_1, Q_2 \subseteq \mathbb{N}$ are recursively enumerable, is $Q_1 \setminus Q_2$ recursively enumerable?
(iii) If $Q \subseteq \mathbb{N}$ is recursively enumerable and $A \subseteq \mathbb{N}$ is recursive, then $Q \setminus A$ is recursively enumerable.
(iv) Show that there are infinitely many recursively enumerable subsets of $\mathbb{N}$ which are not recursive.

Exercise 43. (i) Suppose that $R \subseteq \mathbb{N}$ is recursively enumerable and $k \in \mathbb{N}$. Find a recursive function $g_{R,k} : \mathbb{N} \to \{0, 1\}$ such that

$$\forall_n(g_{R,k}(n) = 0) \leftrightarrow k \notin R.$$

(ii) Suppose that $g : \mathbb{N} \to \{0, 1\}$ is recursive. Then, there is no decision procedure to show

$$\forall_n(g(n) = 0) \ \vee \ \exists_n(g(n) = 1).$$

16. The Ackermann function

Aufgabe 44. Sei $A : \mathbb{N}^2 \to \mathbb{N}$ definiert durch

$$A(0, y) = y + 1$$

$$A(x + 1, 0) = A(x, 1)$$

$$A(x + 1, y + 1) = A(x, A(x + 1, y)).$$

(a) Zeigen Sei die folgende Eigenschaften von A:
(i) $A(x, y) > y$
(ii) $A(x, y) < A(x, y + 1)$

(iii) $y < z \to A(x, y) < A(x, z)$
(iv) $A(x + 1, y) \geq A(x, y + 1)$
(v) $A(x, y) > x$
(vi) $A(x, y) < A(x + 1, y)$
(vii) $y < z \to A(y, x) < A(z, x)$
(b) Wenn $r \geq 0$ und $f : \mathbb{N}^k \to \mathbb{N}$, ist f höchstens vom Rang r genau dann wenn

$$\forall \vec{m}\ f(\vec{m}) \leq A(r, \max(\vec{m})).$$

Man zeige:
(i) Die konstante Funktion m ist höchstens vom Rang m.
(ii) Die Fuktion I_i^k ist höchstens vom Rang 0.
(iii) Die Funktion $\mathrm{Succ}(x) = x + 1$ ist höchstens vom Rang 1.
(iv) Sei h definiert durch

$$h(x_1, \ldots, x_m) = g(f_1(x_1, \ldots, x_m), \ldots, f_k(x_1, \ldots, x_m)).$$

Wenn g höchstens vom Rang s ist und f_i höchstens vom Rang r_i ist, für jeden i, dann ist h höchstens vom Rang $\max\{r_1, \ldots, r_k, s\} + 2$.
(v) Sei h definiert durch

$$h(0) = b$$

$$h(n + 1) = f(h(n)).$$

Wenn f höchstens vom Rang s ist, dann ist h höchstens vom Rang $s + b + 1$.

References

[1] H. D. Ebbinghaus, J. Flum, and W. Thomas: *Mathematical Logic*, Springer-Verlag, 1984.

[2] H. B. Enderton: *A Mathematical Introduction to Logic*, Academic Press, 1972.

[3] J. D. Monk: *Mathematical Logic*, Springer-Verlag, 1976.

[4] A.S. Troelstra and H. Schwichtenberg: *Basic Proof Theory*, 2nd edition, Cambridge, 2000.