Highlights in Logic
Block 1: First–Order Logic FOL

— Preliminary Version —

J. Schmid, M. Sprenger

March 2, 2008

Contents

1 Introduction

Mathematical Logic is concerned with how constructs in formal languages relate to each other and to mathematical objects. These formal constructs are—as is the case for natural languages—built from atomic symbols (e.g. letters) to form increasingly complex entities (words, sentences, paragraphs). The realm of syntax deals with the rules of formation for these more complex constructs. Once we attach meaning to the constructs, we step over to the domain of semantics.

Formal languages in Mathematics in general and Mathematical Logic are no different: We distinguish between syntax and semantics. Of course, as may be expected from a discipline such as Mathematics, the rules governing the construction of syntactical objects and those of their interpretation are more rigorous than the corresponding rules for natural languages.

Another way of looking at Mathematical Logic is by saying that it investigates the possibilities of formalizing the notion of truth and the way it can or cannot be grasped algorithmically. Examples of such algorithmical approximations of truth in Mathematical contexts are proofs: Proofs settle once and for all the question whether a mathematical statement is true or false. Since the notion of proof is—even in mathematical contexts—still a somewhat vague concept, we need to formalize a stricter notion, which will serve as our formal counterpart. We call these formalizations of proofs *deductions*. Being formalizations, deductions obey specific rules. The exact form of these rules is not fixed. There are various different ways to define the notion of deductions and still achieving the same goals: Completeness, Correctness, etc. By collecting all the well–behaved systems of deductions under the notion of *adequate Hilbert–systems*, we take this fact into account.

While syntax deals with *deducibility*, semantics is concerned with *truth*. We will define structures, i.e. potential models of sets of first–order sentences, and we will also define rigorously what is meant by "being true in a structure". It is clear that both the syntactic as well as the semantic side with its strict definition of truth constitute a restriction on the kind of concepts we can consider. Still, as the Completeness Theorem shows, the formalism of deductions results in a considerable expressive power if we insist on the Löwenheim–Skolem Property to hold for our Logic. Unfortunately, this result and the concepts related to it lie beyond the realm of First–Order Logic and its Model Theory. Only in the last Section 9 will we take a quick look at the field concerned with these questions, the so–called *Abstract Model Theory*.

2 Basic Definitions

Before we can treat deductions as mathematical objects we have to introduce formal languages. Once we have formulae at our disposal we can consider finite sequences of them. Thus, the framework of this section allows us to define proofs formally as specific finite sequences of formulae.

2.1 Formal Languages, Terms and Formulae

All the constructs defined in this sections are strings, i.e. finite sequences of elements taken from a set of *symbols*, the *alphabet*.

An alphabet defines which symbols can be used to form terms, formulae and finally also deductions. The mathematical objects we are going to investigate are determined by the alphabet and the rules which specify how to combine symbols in order to form terms and formulae. The entirety of objects pertaining to an alphabet is denoted by the term *formal language*.

Basically, formal languages are characterized by their *non–logical symbols*. All other symbols (*logical* and *auxiliary* symbols) are common to any formal language of first–order logic.

Definition 2.1. The *alphabet* of a formal language $\mathcal{L}$ (of first–order logic) consists of the following symbols:

Connectives	$\neg, \wedge$	*logical symbols*
Quantifier	$\forall$	*logical symbols*
The equality symbol	$\doteq$	*logical symbols*
Variable symbols	$v_n, n \in \mathbb{N}$	*logical symbols*
Relation symbols	$R_i \quad (i \in I)$	*non-logical symbols*
Function symbols	$f_j \quad (j \in J)$	*non-logical symbols*
Constant symbols	$c_k \quad (k \in K)$	*non-logical symbols*
Auxiliary symbols	parentheses, comma	

Every alphabet of a first-order language is infinite since there are infinitely many variable symbols $v_n, n \in \mathbb{N}$. More precisely, the cardinality of an alphabet with index-sets I, J and K is $max\{\aleph_0, card(I), card(J), card(K)\}$.

Definition 2.2. A *formal (first-order) language (with equality)* $\mathcal{L}$ is characterized by

- the alphabet of the language,
- three index sets I, J, K,
- two arity functions $\lambda : I \longrightarrow \mathbb{N} \setminus \{0\}$ and $\mu : J \longrightarrow \mathbb{N} \setminus \{0\}$, where $\lambda(i)$ is the arity of the i-th relation symbol R_i and $\mu(j)$ the arity of the j-th function symbol f_j.

Since the only relevant aspect of the non–logical symbols lies in the index sets I, J, K and the arity functions λ and μ, any two formal languages with identical index sets and arity functions are equivalent, i.e. there is a bijective correspondence between them which preserves all constructs and definitions. This observation and the fact that I and J are implicitly given as the domains of the arity functions λ and μ respectively, justifies the notation $\mathcal{L} = \langle \lambda, \mu, K \rangle$, i.e. a formal language $\mathcal{L}$ is completely determined by the triple $\langle \lambda, \mu, K \rangle$.

Note that all formal languages share the same supply of variables, therefore we can set

$$Vbl := Vbl\,\mathcal{L} := \{v_0, v_1, v_2, \ldots\}.$$

In accordance with the remark following Definition 2.1, we define the *cardinality of a formal language* $\mathcal{L} = \langle \lambda, \mu, K \rangle$ to be given by

$$card\,\mathcal{L} = max\{\aleph_0, card\,I, card\,J, card\,K\}.$$

Definition 2.3 (Terms). Let $\mathcal{L} = \langle \lambda, \mu, \mathcal{K} \rangle$ be a formal language. Then a finite string t of symbols from $\mathcal{L}$ is an $\mathcal{L}$-*term* if either

1. $t \in Vbl$,
2. t is a constant symbol c_k, $k \in K$.
3. t is $f_j(t_1, t_2, \ldots, t_{\mu(j)})$ for $\mathcal{L}$-terms $t_1, t_2, \ldots, t_{\mu(j)}, j \in J$.

The set of all $\mathcal{L}$-terms is denoted by $Trm\,\mathcal{L}$. If the formal language is obvious from the given context we simply speak of a term instead of an $\mathcal{L}$-term.

We note that the set of $\mathcal{L}$-terms has the same cardinality as the alphabet of $\mathcal{L}$.

Definition 2.4 (Formulae). Let $\mathcal{L} = \langle \lambda, \mu, \mathcal{K} \rangle$ be a formal language. Then a finite string φ of symbols from $\mathcal{L}$ is an *$\mathcal{L}$-formula* if either

1. φ is $t_1 \doteq t_2$ for $\mathcal{L}$-terms t_1, t_2,
2. φ is $R_i(t_1, t_2, \ldots, t_{\lambda(i)})$ for $\mathcal{L}$-terms $t_1, t_2, \ldots, t_{\lambda(i)}, i \in I$,
3. φ is $(\neg\alpha)$ for an $\mathcal{L}$-formula α,
4. φ is $(\alpha \wedge \beta)$ for $\mathcal{L}$-formulae α and β,
5. φ is $(\forall v\alpha)$ for an $\mathcal{L}$-formula α and any variable v.

The set of all $\mathcal{L}$–formulae is denoted by $Fml\,\mathcal{L}$.

If the formal language is clear from the given context we simply speak of a formula instead of an $\mathcal{L}$-formula.

Formulae that satisfy one of the first two clauses are called *atomic formulae*.

In Definition 2.1 the symbol $\forall$ was denoted as a quantifier. In a formula $\forall v_n\alpha$ we use the term "quantifier" to denote both $\forall$ and $\forall v_n$.

Again, we note that both the set of $\mathcal{L}$-formulae and the set of atomic $\mathcal{L}$-formulae have the same cardinality as the alphabet of $\mathcal{L}$.

The following abbreviations are standard and introduced for convenience:

Notation 2.5. For all $\varphi, \psi \in Fml\,\mathcal{L}$ and all variables v,

1. $(\varphi \vee \psi)$ stands for $(\neg((\neg\varphi) \wedge (\neg\psi)))$,
2. $(\varphi \rightarrow \psi)$ stands for $((\neg\varphi) \vee \psi)$,
3. $(\exists v_n\varphi)$ stands for $(\neg(\forall v_n(\neg\varphi)))$.

Brackets will be omitted to increase legibility as long as the unique reading is guaranteed, i.e. we write $\neg\forall v_n\neg\varphi$ instead of $(\neg(\forall v_n(\neg\varphi)))$.

Definition 2.6. Let v be a variable and φ a formula. An occurrence of v in φ is said to be *free* if one of the following holds:

1. φ is atomic,
2. φ is $\neg\alpha$ and the occurrence of v in α is free,

3. φ is $\alpha \wedge \beta$ and the occurrence of v in α or in β is free,

4. φ is $\forall u\alpha$, v is different from u and the occurrence of v in α is free.

Put $Fr\,\varphi := \{v\,;\, v \text{ occurs freely in } \varphi\}$

Definition 2.7. A formula φ is called a *sentence* if $Fr\,\varphi = \emptyset$.

$Sen\,\mathcal{L} := \{\varphi \in \mathrm{Fml}(\mathcal{L})\,;\, \varphi \text{ is a sentence}\}$.

Notation 2.8. Unless stated otherwise, we let $x, y, z \ldots$ stand for variables, $s, t \ldots$ for terms, $\varphi, \psi \ldots$ for formulae, and $\alpha, \beta, \ldots$ for sentences. Capital Greek letters $\Sigma, \Theta, \ldots$ are reserved for *sets of* sentences or formulae.

We end this section with a definition introducing the universal closure $\forall\varphi$ of a formula φ. $\forall\varphi$ is a sentence obtained from φ by prefixing φ with a quantifier $\forall v$ for every $v \in Fr(\varphi)$, thus binding all free variables of φ.

Definition 2.9 (Universal Closure). Let φ be a formula with $Fr(\varphi) = \{v_1, \ldots, v_n\}$. Then the formula[1] $\forall v_1 \ldots \forall v_n \varphi$, denoted by $\forall\varphi$, is called the universal closure of φ.

2.2 Semantics

Mathematical reasoning in the ordinary sense is the study of the properties of a mathematical structure. Therefore, we must now define the notion of a mathematical structure pertaining to a formal language. That is, we define the notion of $\mathcal{L}$-*structure* for any given first-order language $\mathcal{L}$.

Definition 2.10. Let $\mathcal{L} = \langle \lambda, \mu, K \rangle$ be a formal language. An $\mathcal{L}$-*structure* $\mathcal{A}$ consists of

(i) a non-empty set $A = |\mathcal{A}|$, the *universe* of $\mathcal{A}$,

(ii) for every $i \in I$ a $\lambda(i)$-ary relation $R_i^{\mathcal{A}} \subseteq A^{\lambda(i)}$on A,

(iii) for every $j \in J$ a $\mu(j)$-ary function $f_j^{\mathcal{A}} : A^{\mu(j)} \to A$ on A, and

[1] Closer inspection of this definition reveals that "the" universal closure of a formula φ is *uniquely* defined if and only if φ has at most one free variable. To be accurate, we should rather speak of *an* universal closure, turning the notion into a relation between instead of an operation on formulae. Nevertheless we will stick to this "abus de language", since for all our future purposes the different universal closures are easily seen to be equivalent.

(iv) for every $k \in K$ a fixed element (constant) $c_k^{\mathcal{A}} \in A$.

$R_i^{\mathcal{A}}$, $f_j^{\mathcal{A}}$ and $c_k^{\mathcal{A}}$ are called the *interpretations* of R_i, f_j and c_k in $\mathcal{A}$.

The next definition formalizes the relationship between the object a term refers to and the interpretation of the variables occurring in this term.

Definition 2.11. Let $\mathcal{L}$ be a formal language and $\mathcal{A}$ an $\mathcal{L}$-structure.

(i) A function $h : Vbl(\mathcal{L}) \to |\mathcal{A}|$ is called a *variable assignment into* $\mathcal{A}$.

(ii) If h is an assignment function, v a variable and $a \in |\mathcal{A}|$, then the assignment function $h\binom{v}{a}(u)$ given by

$$h\binom{v}{a}(u) = \begin{cases} h(u) & u \neq v \\ a & u = v \end{cases}$$

is called a *v-modification of the assignment function h.*

We are now in a position to formally define the interpretation of a term t and the truth of a formula in an $\mathcal{L}$-structure.

Definition 2.12. Let $\mathcal{L}$ be a formal language, $\mathcal{A}$ an $\mathcal{L}$-structure and h a variable assignment function into $\mathcal{A}$. Then the function $\bar{h} : Trm(\mathcal{L}) \to |\mathcal{A}|$, called the *term assignment function generated by* h is defined inductively by assigning to a term t

1. $h(v)$ if t is the variable v,
2. $c_k^{\mathcal{A}}$ if t is the constant c_k,
3. $f_j^{\mathcal{A}}(\bar{h}(t_1), \ldots, \bar{h}(t_{\mu(j)}))$ if t is $f_j(t_1, \ldots, t_{\mu(j)})$.

Definition 2.13. Let $\mathcal{L}$ be a formal language, φ an $\mathcal{L}$-formula, $\mathcal{A}$ an $\mathcal{L}$-structure and h a variable assignment into $\mathcal{A}$. Then we say $\mathcal{A}$ *satisfies* φ *under the assignment* h, denoted by $\mathcal{A} \models \varphi[h]$, if and only if

1. φ is $t_1 \doteq t_2$ for $\mathcal{L}$-terms t_1, t_2, and $\bar{h}(t_1) = \bar{h}(t_2)$,
2. φ is $R_i(t_1, \ldots, t_{\lambda(i)})$ for $\mathcal{L}$-terms $t_1, \ldots, t_{\lambda(i)}$, and $(\bar{h}(t_1), \ldots, \bar{h}(t_{\lambda(i)})) \in R_i^{\mathcal{A}}$,

3. φ is $\neg\alpha$ for an $\mathcal{L}$-formula α, and $\mathcal{A} \not\models \alpha[h]$ ("not $\mathcal{A} \models \alpha[h]$"),

4. φ is $\alpha \wedge \beta$ for $\mathcal{L}$-formulae α and β, and both $\mathcal{A} \models \alpha[h]$ and $\mathcal{A} \models \beta[h]$,

5. φ is $\forall v\alpha$ for an $\mathcal{L}$-formula α and a variable v, and $\mathcal{A} \models \alpha[h\binom{v}{a}]$ for all $a \in |\mathcal{A}|$.

If Σ is a set of $\mathcal{L}$-formulae, we say $\mathcal{A}$ satisfies Σ with assignment h if $\mathcal{A} \models \sigma[h]$ for all $\sigma \in \Sigma$. Notation: $\mathcal{A} \models \Sigma[h]$.

φ is said to be *satisfiable* if there is an $\mathcal{L}$-structure $\mathcal{A}$ and an assignment h such that $\mathcal{A} \models \varphi[h]$.

It is easily seen that only the values of the assignment function h on the free variables of φ matter for the truth of $\mathcal{A} \models \varphi[h]$.

Some additional notions need to be introduced:

Definition 2.14 (Validity). Let φ be an $\mathcal{L}$-formula and $\mathcal{A}$ an $\mathcal{L}$-structure. Then

1. φ is said to be *valid* or *true in* $\mathcal{A}$, denoted by $\mathcal{A} \models \varphi$, if $\mathcal{A} \models \varphi[h]$ holds for all variable assignments $h : Vbl \to |\mathcal{A}|$, in which case $\mathcal{A}$ is called a *model of* φ.

2. If $\mathcal{B} \models \varphi$ for every $\mathcal{L}$-structure $\mathcal{B}$ then φ is said to be *valid*.

3. If $\Sigma \subset Fml(\mathcal{L})$ then $\mathcal{A}$ is said to be a *model* of Σ, denoted by $\mathcal{A} \models \Sigma$, if $\mathcal{A} \models \varphi$ holds for all $\varphi \in \Sigma$.

The next lemma shows that—with respect to validity—a formula φ and its universal closure $\forall\varphi$ are equivalent:

Lemma 2.15. *Let $\mathcal{A}$ be an $\mathcal{L}$-structure and $\varphi \in Fml(\mathcal{L})$. Then the following holds:*

$$\mathcal{A} \models \varphi \qquad \textit{iff} \qquad \mathcal{A} \models \forall\varphi.$$

Now we come to a definition whose full importance will only become clear later. It captures the notion of implication with regard to structures, to be called *logical implication*.

Definition 2.16 (Logical Implication). Let $\mathcal{L}$ be a formal language, Σ a set of $\mathcal{L}$-formulae and φ an $\mathcal{L}$-formula.

If for all $\mathcal{L}$-structures $\mathcal{A}$ we have that $\mathcal{A} \models \Sigma$ implies $\mathcal{A} \models \varphi$, then we say that Σ *logically implies* φ and write

$$\Sigma \Vdash \varphi.$$

The close resemblance between $\vdash$ and $\Vdash$ is not a coincidence. The intention behind this symbolism will become clear once we introduced the notion of *(Gödel)–Completeness* for Hilbert–systems (cf. Section 5.2). Some authors use $\models$ instead of $\Vdash$ and accept the resulting ambiguity, since it proves to be no problem in the presence of the above–mentioned completeness.

2.3 Structures: Basic Notions and Results

Definition 2.17 (Congruence). Let $\mathcal{L} = \langle \lambda, \mu, K \rangle$ be a formal language and $\mathcal{A}$ an $\mathcal{L}$–structure. An equivalence relation $\sim \subseteq |\mathcal{A}| \times |\mathcal{A}|$ is called an $\mathcal{L}$–*congruence* on $\mathcal{A}$ if $\sim$ is *compatible* with the interpretations of the function– and relation symbols of $\mathcal{L}$ in the following sense:

$$a_1 \sim_\Sigma a'_1, \ldots, a_{\mu(j)} \sim_\Sigma a'_{\mu(j)} \text{ implies } f_j^{\mathcal{A}}(a_1, \ldots, a_{\mu(j)}) \sim_\Sigma f_j^{\mathcal{A}}(a'_1, \ldots, a'_{\mu(j)})$$

and

$$a_1 \sim a'_1, \ldots, a_{\lambda(i)} \sim a'_{\lambda(i)} \text{ implies } [R_i^{\mathcal{A}}(a_1, \ldots, a_{\lambda(i)}) \text{ iff } R_i^{\mathcal{A}}(a'_1, \ldots, a'_{\lambda(i)})]$$

for all $j \in J, i \in I$ and $a_k, a'_k \in |\mathcal{A}|$. As is the custom for equivalence relations, we let $[a]_\sim$ denote the $\sim$–class of $a \in |\mathcal{A}|$, i.e. $[a]_\sim = \{b \in |\mathcal{A}|\,;\, b \sim a\}$.

Proposition 2.18. *Let $\mathcal{L} = \langle \lambda, \mu, K \rangle$ be a formal language. If $\sim$ is an $\mathcal{L}$–congruence on the $\mathcal{L}$–structure $\mathcal{A}$, then the* quotient *$\mathcal{A}/\sim$ is again an $\mathcal{L}$–structure, where $\mathcal{A}/\sim$ is given by:*

(i) $|\mathcal{A}/\sim| = \{[a]_\sim\,;\, a \in |\mathcal{A}|\}$;

(ii) $f_j^{\mathcal{A}/\sim}([a_1]_\sim, \ldots, [a_{\mu(j)}]_\sim) = [f_j^{\mathcal{A}}(a_1, \ldots, a_{\mu(j)})]_\sim$;

(iii) $R_i^{\mathcal{A}/\sim}([a_1]_\sim, \ldots, [a_{\lambda(i)}]_\sim)$ *iff* $R_i^{\mathcal{A}}(a_1, \ldots, a_{\mu(j)})$;

(iv) $c_k^{\mathcal{A}/\sim} = [c_k^{\mathcal{A}}]_\sim$.

Especially, clauses (ii) and (iii) are well-defined.

Definition 2.19. Let $\mathcal{L} = \langle \lambda, \mu, K \rangle$ be a formal language and $\mathcal{A}, \mathcal{B}$ be $\mathcal{L}$–structures. Then:

1. $\mathcal{A}$ is an $\mathcal{L}$*–substructure* of $\mathcal{B}$, notation $\mathcal{A} \subseteq_{\mathcal{L}} \mathcal{B}$, if the following conditions are met:

   (i) $|\mathcal{A}| \subseteq |\mathcal{B}|$,

   (ii) $f_j^{\mathcal{A}} = f_j^{\mathcal{B}} \mid_{|\mathcal{A}|^{\mu(j)}}$,

   (iii) $R_i^{\mathcal{A}} = R_i^{\mathcal{B}} \cap |\mathcal{A}|^{\mu(j)}$,

   (iv) $c_k^{\mathcal{A}} = c_k^{\mathcal{B}}$,

   for all $j \in J, i \in I, k \in K$.

2. $\mathcal{A}$ is an $\mathcal{L}$*–elementary substructure* of $\mathcal{B}$, if $\mathcal{A} \subseteq \mathcal{B}$ and for all $\mathcal{L}$–formulae φ with free variables $x_1, \ldots, x_n$, all $a_1, \ldots, a_n \in |\mathcal{A}|$ and all valuations h,

$$\mathcal{A} \models \varphi[h\binom{x_1}{a_1} \cdots \binom{x_n}{a_n}] \qquad \text{iff} \qquad \mathcal{B} \models \varphi[h\binom{x_1}{a_1} \cdots \binom{x_n}{a_n}].$$

   Notation: $\mathcal{A} \preceq_{\mathcal{L}} \mathcal{B}$.

Note that substructures are closed under the interpretations of the function symbols and contain all interpretations of the constant symbols of the language $\mathcal{L}$.

Definition 2.20. Let $\mathcal{L} = \langle \lambda, \mu, K \rangle$ be a formal language and $\mathcal{A}, \mathcal{B}$ be $\mathcal{L}$–structures. A mapping $\eta : |\mathcal{A}| \longrightarrow |\mathcal{B}|$ is called an $\mathcal{L}$–isomorphism from $\mathcal{A}$ to $\mathcal{B}$ provided that

(i) η is a bijection;

(ii) $\eta(f_j^{\mathcal{A}}(a_1, \ldots, a_{\mu(j)})) = f_j^{\mathcal{B}}(\eta(a_1), \ldots, \eta(a_{\mu(j)}))$;

(iii) $R_i^{\mathcal{A}}(a_1, \ldots, a_{\lambda(i)})$ iff $R_i^{\mathcal{B}}(\eta(a_1), \ldots, \eta(a_{\lambda(i)}))$;

(iv) $\eta(c_k^{\mathcal{A}}) = c_k^{\mathcal{B}}$, for all $j \in J, i \in I, k \in K$. If there is an $\mathcal{L}$–isomorphism from $\mathcal{A}$ to $\mathcal{B}$, then $\mathcal{A}$ to $\mathcal{B}$ are said to be $\mathcal{L}$*–isomorphic*, notation $\mathcal{A} \cong_{\mathcal{L}} \mathcal{B}$.

Proposition 2.21. *If $\mathcal{L}$ is a formal language, $\mathcal{A}, \mathcal{B}$ $\mathcal{L}$–structures and η an $\mathcal{L}$–isomorphism from $\mathcal{A}$ to $\mathcal{B}$, then for any $\mathcal{L}$–formula φ and any variable assignment h into $\mathcal{A}$,*

$$\mathcal{A} \models \varphi[h] \qquad \textit{iff} \qquad \mathcal{B} \models \varphi[\eta \circ h].$$

Definition 2.22. Let $\mathcal{L} = \langle \lambda, \mu, K \rangle$ be a formal language and $\mathcal{A}, \mathcal{B}$ be $\mathcal{L}$–structures. Then $\mathcal{A}$ and $\mathcal{B}$ are $\mathcal{L}$*–elementary equivalent*, notation $\mathcal{A} \equiv_{\mathcal{L}} \mathcal{B}$, if for all $\mathcal{L}$–sentences α,

$$\mathcal{A} \models \alpha \qquad \text{iff} \qquad \mathcal{B} \models \alpha.$$

The notions are related as shown in the following proposition:

Proposition 2.23. *For a formal language $\mathcal{L}$ and $\mathcal{A}, \mathcal{B} \in Str\, \mathcal{L}$,*

(i) $\mathcal{A} \preceq_{\mathcal{L}} \mathcal{B} \quad \overset{\Longrightarrow}{\not\Longleftarrow} \quad \mathcal{A} \subseteq_{\mathcal{L}} \mathcal{B}$;

(ii) $\mathcal{A} \preceq_{\mathcal{L}} \mathcal{B} \quad \overset{\Longrightarrow}{\not\Longleftarrow} \quad \mathcal{A} \equiv_{\mathcal{L}} \mathcal{B}$;

(iii) $\mathcal{A} \cong_{\mathcal{L}} \mathcal{B} \quad \overset{\Longrightarrow}{\not\Longleftarrow} \quad \mathcal{A} \equiv_{\mathcal{L}} \mathcal{B}$.

Sketch of Proof:

(i) $\Longrightarrow$: By definition.
$\not\Longleftarrow$: Consider $\mathcal{A} = \langle \mathbb{N}, < \rangle$, $\mathcal{B} = \langle \mathbb{N} \setminus \{0\}, < \rangle$ and $\exists v_1 v_1 < v_2$.

(ii) $\Longrightarrow$: Trivial, since sentences are formulae.
$\not\Longleftarrow$: $\mathcal{A} \equiv_{\mathcal{L}} \mathcal{B}$ does not imply $|\mathcal{A}| \subseteq |\mathcal{B}|$.

(iii) $\Longrightarrow$: Using structural induction over terms and formulae, it can be shown that isomorphisms preserve interpretations, term assignments and validity of formulae.
$\not\Longleftarrow$: Using advanced methods (cf. Section 6.3), it can be shown that $\langle \mathbb{Q}, \leq \rangle \equiv \langle \mathbb{R}, \leq \rangle$. But $\langle \mathbb{Q}, \leq \rangle \not\cong \langle \mathbb{R}, \leq \rangle$. (cardinality!)

■

3 Deductions

Deductions are finite sequences of formulae obeying certain laws. To be just a little more specific, we will allow formulae in a deduction to be either formulae from a given set or formulae which can be formed from formulae with lower indices using certain rules. The exact nature of these rules may vary, which is the reason we do not only introduce one system of deduction but a whole class, among which we will distinguish those systems which are suitable for our purposes.

3.1 Hilbert–Systems

Using deductions, we *deduce* formulae which are either to be regarded as true (or given without any premises), or which are the result of applying a rule which preserves truth to formulae which are known to be true since they already appear in the deduction. Abstracting these thoughts, we get the following general definition:

Definition 3.1 (Hilbert–Systems). Let $\mathcal{L}$ be a formal language. A *Hilbert–style System* $\mathcal{H}$ *of Formal Deductions*, or a *Hilbert–system* for short, is a pair $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$, where

1. $\mathcal{L}$ is a formal language and
2. $\mathcal{R} \subseteq \{\langle \Sigma, \varphi \rangle \,;\, \Sigma \subseteq Fml\,\mathcal{L} \text{ finite}, \varphi \in Fml\,\mathcal{L}\}$.

The elements of $\mathcal{R}$ are called *rules (of deduction)* of $\mathcal{H}$. The first component of the rules is called the *set of premises*, while the second component is called the *conclusion*. Conclusions of rules with an empty set of premises are called *(logical) axioms* of $\mathcal{H}$.

The idea behind this formalism will become clearer once we introduce "our" specimen of formal systems $\mathcal{H}_{\mathcal{L}}$. However, to link to the introductory remarks of this section, we observe that axioms are regarded as true without constraints and can therefore be inserted anywhere in a deduction, whereas the rules of deductions rely on the fact that the premises (i.e. the formulae in the first component of the rule) have already been deduced in order to allow the conclusion (i.e. the second part of the rule) to be inserted to a deduction.

Definition 3.2 (Hilbert–Style Formal Deduction). Let $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ be a Hilbert–system. Let $\sigma \subseteq Fml\,\mathcal{L}$ and $\varphi \in Fml\,\mathcal{L}$. A finite sequence $\psi_0, \ldots, \psi_n$ of $\mathcal{L}$–formulae is then called a *(Hilbert–style) formal deduction of* φ *from* Σ *in* $\mathcal{H}$ if $\psi_n = \varphi$ and for all $i \in \{0, \ldots, n-1\}$, at least one of the following conditions is satisfied:

1. $\psi_i \in \Sigma$;
2. for some $j_1, \ldots, j_m \in \{0, \ldots, i-1\}$, $\langle \{\psi_{j_1}, \ldots, \psi_{j_m}\}, \psi_i \rangle \in \mathcal{R}$.

We let $\Sigma \vdash_{\mathcal{H}} \varphi$ stand for the fact that there is a formal deduction of φ from Σ in $\mathcal{H}$. If not, we write $\Sigma \nvdash_{\mathcal{H}} \varphi$. If $\Sigma = \emptyset$, we write $\vdash_{\mathcal{H}} \varphi$ instead of $\emptyset \vdash_{\mathcal{H}} \varphi$.

As usual, subscripts may be dropped if there is no danger of ambiguity regarding the formal system under consideration.

3.2 A Concrete Hilbert–System

For every formal language $\mathcal{L}$ we specify a Hilbert–system $\mathcal{H}_{\mathcal{L}} = \langle \mathcal{L}, \mathcal{R}_{\mathcal{L}} \rangle$ which will serve as "testing ground" for for Hilbert–systems when it comes to criteria such as completeness and correctness. Therefore we must define the set of rules $\mathcal{R}_{\mathcal{L}}$ for these systems in dependence of the language $\mathcal{L}$.

Some of the simplest true statements in natural languages are those whose truth is verified by a simple syntactical analysis: the so–called tautologies. Since tautologies may also appear in (meta–)mathematical contexts, we will now formalize those syntactical truths and incorporate them into $\mathcal{H}_{\mathcal{L}}$.

Definition 3.3 (Truth–Values and –Functions). For what follows, assume $\mathcal{T} = \{t, f\}$ is a fixed set with two elements t, f, $t \neq f$, the so–called *truth–values* (*true, false*). Moreover, on $\mathcal{T}$ we define the following *truth–functions*:

- $\sim: \mathcal{T} \longrightarrow \mathcal{T}$, given by

$$\sim t = f \text{ and } \sim f = t.$$

- $\star : \mathcal{T} \times \mathcal{T} \longrightarrow \mathcal{T}$, given by

$$t \star t = t \qquad \text{and} \qquad t \star f = f \star t = f \star f = f.$$

Definition 3.4 (Propositional Valuations).

1. A *propositional valuation* for the formal language $\mathcal{L}$ is a function p assigning values in $\mathcal{T}$ to all atomic $\mathcal{L}$–formulae.

2. If p is a propositional valuation for $\mathcal{L}$, then the *propositional value* of a (not necessarily atomic) $\mathcal{L}$–formula φ under p, notation $\overline{p}(\varphi)$, is defined inductively as follows:

   - If φ is atomic, then $\overline{p}(\varphi) = p(\varphi)$.
   - If $\varphi = \neg\psi$, then $\overline{p}(\varphi) = \sim \overline{p}(\psi)$.
   - If $\varphi = \psi_1 \wedge \psi_2$, then $\overline{p}(\varphi) = \overline{p}(\psi_1) \star \overline{p}(\psi_2)$.

3. An $\mathcal{L}$–formula φ is called a *propositional tautology* if $\overline{p}(\varphi) = t$ for all propositional valuations p.

Since for a given formula φ, the propositional values of φ under valuations p_1 and p_2 can only differ if p_1 and p_2 have different values for at least one atomic formula occurring in φ, and since formulae are finite entities, the question whether a given formula is a propositional tautology can always be answered, i.e. it is decidable if the formula is a propositional tautology.

In order to define $\mathcal{H}_{\mathcal{L}}$, we need to introduce the technical notions of *substitution* and of a variable *being free for* a term in a formula:

Definition 3.5. Let φ be an $\mathcal{L}$-formula, t an $\mathcal{L}$-term and v a variable. The $\mathcal{L}$-formula $\varphi(v/t)$ ("φ with v replaced by t") is defined inductively as follows:

1. If φ is $t_1 \doteq t_2$ for $\mathcal{L}$-terms t_1, t_2 then $\varphi(v/t)$ is $t_1(v/t) \doteq t_2(v/t)$.

2. If φ is $R_i(t_1, \ldots, t_{\lambda(i)})$ is for $\mathcal{L}$-terms $t_1, \ldots, t_{\lambda(i)}$ then $\varphi(v/t)$ is $R_i(t_1(v/t), \ldots, t_{\lambda(i)}(v/t))$.

3. If φ is $\neg\alpha$ for an $\mathcal{L}$-formula α then $\varphi(v/t)$ is $\neg\alpha(v/t)$.

4. If φ is $\alpha \wedge \beta$ for $\mathcal{L}$-formulae α and β then $\varphi(v/t)$ is $\alpha(v/t) \wedge \beta(v/t)$.

5. If φ is $\forall u\alpha$ for an $\mathcal{L}$-formula α and a variable u then

$$\varphi(v/t) = \begin{cases} \forall u\alpha(v/t) & v \neq u \\ \varphi & v = u. \end{cases}$$

Definition 3.6. Let φ be an $\mathcal{L}$-formula, t a term and v a variable. Then t *is free for v in φ* if

1. φ is atomic;
2. φ is $\neg\alpha$ and t is free for v in α;
3. φ is $\alpha \wedge \beta$ and t is free for v in both α and β;
4. φ is $\forall x \alpha$ and either
   (a) v is not free in α, or
   (b) x does not occur in t and t is free for v in α.

Definition 3.7 ($\mathcal{R}_\mathcal{L}$). Let $\mathcal{L}$ be a formal language. First, define the following sets of formulae:

(i) Let $T_\mathcal{L}$ be the set of all propositional tautologies for $\mathcal{L}$;

(ii) $I := \{v_0 \doteq v_0, v_1 \doteq v_2 \rightarrow (v_1 \doteq v_3 \rightarrow v_2 \doteq v_3)\}$;

(iii) $Rel_\mathcal{L} :=$
$\{(v_1 \doteq v_1' \wedge \ldots \wedge v_{\lambda(i)} \doteq v_{\lambda(i)}') \rightarrow (R_i(v_1, \ldots, v_{\lambda(i)}) \rightarrow R_i(v_1', \ldots, v_{\lambda(i)}'))\,;$
$v_1, \ldots, v_{\lambda(i)}, v_1', \ldots, v_{\lambda(i)}' \in Vbl\}$;

(iv) $Fun_\mathcal{L} :=$
$\{(v_1 \doteq v_1' \wedge \ldots \wedge v_{\mu(j)} \doteq v_{\mu(j)}') \rightarrow f_j(v_1, \ldots, v_{\mu(j)}) \doteq f_j(v_1', \ldots, v_{\mu(j)}')\,;$
$v_1, \ldots, v_{\mu(j)}, v_1', \ldots, v_{\mu(j)}' \in Vbl\}$;

(v) $Q_\mathcal{L} := \{\forall v\, \alpha \rightarrow \alpha(v/t)\,;\, t \in Trm\,\mathcal{L} \text{ free for } v \in Vbl \text{ in } \alpha \in Fml\,\mathcal{L}\}$.

(vi) $M_\mathcal{L} := \{\langle\{\varphi \rightarrow \psi, \varphi\}, \psi\rangle\,;\, \psi, \varphi \in Fml\,\mathcal{L}\}$

(vii) $G_\mathcal{L} := \{\langle\{\varphi \rightarrow \psi\}, \varphi \rightarrow \forall v \psi\rangle\,;\, \varphi, \psi \in Fml\,\mathcal{L}, v \in Vbl, v \notin Fr\,\varphi\}$.

Then we set

$$\mathcal{R}_\mathcal{L} := M_\mathcal{L} \cup G_\mathcal{L} \cup \{\langle\emptyset, \varphi\rangle\,;\, \varphi \in T_\mathcal{L} \cup I \cup Rel_\mathcal{L} \cup Fun_\mathcal{L} \cup Q_\mathcal{L}\}$$

and finally $\mathcal{H}_\mathcal{L} := \langle\mathcal{L}, \mathcal{R}_\mathcal{L}\rangle$.

The rules in $M_\mathcal{L}$ and $G_\mathcal{L}$ are often referred to as instances of *Modus Ponens* and *Generalization*, respectively.

3.3 Properties of Hilbert–Systems

In order to define and verify properties of Hilbert–systems, we must be able to compare Hilbert–systems as follows: Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two Hilbert–systems for the same formal language $\mathcal{L}$. We write $\mathcal{H}_1 \leq \mathcal{H}_2$ if

$$\Sigma_{\mathcal{H}_1} \vdash \alpha \qquad \text{implies} \qquad \Sigma_{\mathcal{H}_2} \vdash \alpha$$

for all $\Sigma \subseteq Fml\,\mathcal{L}$, $\alpha \in Fml\,\mathcal{L}$.

Note that $\leq$ is a pre–order on the set of all Hilbert–systems for a formal language $\mathcal{L}$, i.e. a transitive and reflexive binary relation. However, $\leq$ is not a partial order since it is *not anti–symmetric*. For our purposes it will be enough to consider Hilbert–systems only modulo the equivalence relation $\sim$ defined by $\mathcal{H}_1 \sim \mathcal{H}_2$ iff $\mathcal{H}_1 \leq \mathcal{H}_2 \leq \mathcal{H}_1$, because

$$(\Sigma_{\mathcal{H}_1} \sim \Sigma_{\mathcal{H}_2}) \qquad \text{iff} \qquad (\Sigma_{\mathcal{H}_1} \vdash \alpha \text{ if and only if } \Sigma_{\mathcal{H}_2} \vdash \alpha).$$

Since we are interested in Hilbert–systems whose deductive power is the same as the one of the appropriate $\mathcal{H}_{\mathcal{L}}$, we come to the following definition:

Definition 3.8. A Hilbert–system $\mathcal{H}$ for a formal language $\mathcal{L}$ is called *adequate* if $\mathcal{H} \sim \mathcal{H}_{\mathcal{L}}$.

In the following, we will often use $\vdash$ to denote deducibility in $\mathcal{H}_{\mathcal{L}}$ (and therefore in any adequate Hilbert–system).

3.4 Consistency

Definition 3.9 ((In)consistent Set of Formulae). Let $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ be a Hilbert–system. A set Σ of $\mathcal{L}$–formulae is called $\mathcal{H}$*–inconsistent* if $\Sigma \vdash_{\mathcal{H}} \alpha \wedge (\neg\alpha)$ for some $\mathcal{L}$–sentence α, and $\mathcal{H}$*–consistent* if Σ is not $\mathcal{H}$–inconsistent.

Note that for Hilbert–systems $\mathcal{H}$ with $\mathcal{H}_{\mathcal{L}} \leq \mathcal{H}$,

$$\Sigma \text{ is } \mathcal{H}\text{–inconsistent} \qquad \text{iff} \qquad \Sigma \vdash_{\mathcal{H}} \varphi \text{ for } \textit{all } \mathcal{L}\text{–formulae } \varphi.$$

This follows from the fact that $\vdash_{\mathcal{H}_{\mathcal{L}}} (\alpha \wedge (\neg\alpha)) \to \varphi$.

The finiteness of deductions is crucial for the proof of Proposition 3.11.

Definition 3.10. Let X be any set. Then $\mathcal{S} \subseteq \mathcal{P}(X)$ is called *directed* if

$$S_1, S_2 \in \mathcal{S} \qquad \text{implies} \qquad S_1 \subseteq S_3 \supseteq S_2 \text{ for some } S_3 \in \mathcal{S}.$$

Proposition 3.11. *Let $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ be a Hilbert–system for the formal language $\mathcal{L}$. Let $\mathcal{S}$ be a directed set of $\mathcal{H}$–consistent sets of $\mathcal{L}$–formulae. Then $\bigcup \mathcal{S}$ is $\mathcal{H}$–consistent.*

Sketch of Proof: If $\bigcup \mathcal{S}$ were $\mathcal{H}$–inconsistent, then there would be a $\mathcal{H}$–deduction of some sentence $\alpha \wedge \neg\alpha$ from $\bigcup \mathcal{S}$. Because deductions are finite, we would then find some $\Sigma_s \in \mathcal{S}$ such that $\Sigma_s \vdash_{\mathcal{H}} \alpha \wedge \neg\alpha$. ■

Theorem 3.12. *Let $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ be a Hilbert–system for the formal language $\mathcal{L}$. Then, for every $\mathcal{H}$–consistent set $\Sigma \subseteq Fml\,\mathcal{L}$ there is a maximally $\mathcal{H}$–consistent set $\Sigma_0 \subseteq Fml\,\mathcal{L}$ extending Σ.*

Sketch of Proof: Consider the set P of all $\mathcal{H}$–consistent sets of $\mathcal{L}$–formulae containing Σ as a subset, ordered by set–inclusion. Let C be a non–empty chain in P. Then $\bigcup C$ is again $\mathcal{H}$–consistent by Proposition 3.11. Hence, every chain C has an upper bound $\bigcup C$ in P, therefore we may apply Zorn's Lemma which yields the existence of a maximal element in P, i.e. a maximally $\mathcal{H}$–consistent superset of Σ. ■

3.5 Theories

Starting from classes of $\mathcal{L}$–structures, we investigate the set of sentences which are valid in all structures in these classes.

Definition 3.13. Let $\mathcal{L}$ be a first–order language and $\Sigma \subseteq Sen\,\mathcal{L}$.

(i) Σ is called an $\mathcal{L}$*–theory* if, for some class $\mathbf{K}$ of $\mathcal{L}$–structures,

$$\Sigma = \{\alpha \in Sen\,\mathcal{L}\,;\, \mathcal{A} \models \alpha \text{ for all } \mathcal{A} \in \mathbf{K}\}.$$

(ii) If in (i), $\mathbf{K} = \{\mathcal{A}\}$ is a singleton class, then Σ is called (a) *complete* (set of $\mathcal{L}$–sentences).

If $\Sigma = \{\alpha \in Sen\,\mathcal{L}\,;\, \mathcal{A} \models \alpha \text{ for all } \mathcal{A} \in \mathbf{K}\}$, we say that Σ is the *theory of* $\mathbf{K}$. If $\mathbf{K} = \mathcal{A}$, we say that Σ is the theory of $\mathcal{A}$.

Clearly, complete sets of sentences are theories. Other than theories, complete sets of sentences are always $\mathcal{H}_{\mathcal{L}}$–consistent.

Proposition 3.14. *Let $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ be an adequate Hilbert–system and Σ an $\mathcal{H}$–consistent $\mathcal{L}$–theory. Then*

$$\Sigma \text{ is complete} \quad \text{iff} \quad \mathcal{A} \equiv_{\mathcal{L}} \mathcal{B} \text{ for any two models } \mathcal{A}, \mathcal{B} \text{ of } \Sigma.$$

While the notions "theory" and "complete theory" do not rely on Hilbert–systems, they become meaningful mostly in the context of adequate systems.

Proposition 3.15. *Let $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ be an adequate Hilbert–system.*

(i) $\Sigma \subseteq Sen\,\mathcal{L}$ *is an* $\mathcal{L}$*–theory iff* Σ *is* $\mathcal{H}$*–deductively closed, i.e. iff* $\Sigma \vdash_{\mathcal{H}} \alpha$ *implies* $\alpha \in \Sigma$ *for all* $\alpha \in Sen\,\mathcal{L}$*.*

(ii) $\Sigma \subseteq Sen\,\mathcal{L}$ *is complete iff* Σ *is a maximally* $\mathcal{H}$*–consistent set of* $\mathcal{L}$*–sentences.*

With the help of Proposition 3.15, we can express the fact that a set of sentences is a theory or complete without using the semantical concept of (classes of) structure(s).

4 Henkin–Structures

Henkin–structures provide a comfortable way of producing models from syntactic constructs, which will be used in the proof of the Completeness Theorem 5.6 for adequate Hilbert–systems.

4.1 Enriching Languages

Definition 4.1. Let $\mathcal{L} = \langle \lambda, \mu, K \rangle$ be a first–order language and $\Sigma \subseteq Sen\,\mathcal{L}$. Then Σ is said to *have witnesses* if for every existential $\mathcal{L}$–sentence $\exists x\varphi \in Sen\,\mathcal{L}$, there is an index $k \in K$ such that $\exists x\varphi \to \varphi(x/c_k) \in \Sigma$.

To have witnesses means that there are enough constant symbols to provide an "example" for all existential sentences.

Definition 4.2. Let $\mathcal{L}_1 = \langle \mu_1, \lambda_1, K_1 \rangle$ and $\mathcal{L}_2 = \langle \mu_2, \lambda_2, K_2 \rangle$ be two first–order languages. Then $\mathcal{L}_2$ is said to be *an extension of* (or to be *extending*) $\mathcal{L}_1$ if $\mu_1 \subseteq \mu_2$, $\lambda_1 \subseteq \lambda_2$ and $K_1 \subseteq K_2$.

Proposition 4.3. *Let $\mathcal{L} = \langle \lambda, \mu, K \rangle$ be a first–order language. If $\Sigma \subseteq Sen\,\mathcal{L}$ is $\mathcal{H}_{\mathcal{L}}$–consistent, then there is an extension $\mathcal{L}'$ of $\mathcal{L}$ and $\Sigma' \subseteq Sen\,\mathcal{L}'$ with $\Sigma \subseteq \Sigma'$ such that*

(i) $card\,\mathcal{L}' = card\,\mathcal{L}$,

(ii) Σ' has witnesses and

(iii) Σ' is $\mathcal{H}_{\mathcal{L}'}$–consistent.

Sketch of Proof: Assume Σ is $\mathcal{H}_{\mathcal{L}}$–consistent. For $n \in \mathbb{N}$, define formal languages $\mathcal{L}_n$ and $\Sigma_n \subseteq Fml\,\mathcal{L}_n$ as follows: $\mathcal{L}_0 = \mathcal{L}$, $\Sigma_0 = \Sigma$, and if $\mathcal{L}_i$ is given by $\mathcal{L}_i = \langle \lambda_i, \mu_i, K_i \rangle$, we set $\mathcal{L}_{i+1} := \langle \lambda_i, \mu_i, K_{i+1} \rangle$ where $K_{i+1} := K_i \cup \{c_{\exists x\varphi}\ ;\ \exists x\varphi \in Fml\,\mathcal{L}_i\}$. (We assume that all the constant symbols are pairwise distinct.) Define Σ_{i+1} by $\Sigma_{i+1} := \Sigma_i \cup \{\exists x\varphi \to \varphi(x/c_{\exists x\varphi})\}$. Finally, we set $\mathcal{L}' := \bigcup_{n\in\mathbb{N}} \mathcal{L}_n$ and $\Sigma' := \bigcup_{n\in\mathbb{N}} \Sigma_n$. Simple calculation shows that $card\,\mathcal{L}' = card\,\mathcal{L}$. Also, by induction we can show that all the Σ_i are $\mathcal{H}_{\mathcal{L}'}$–consistent, hence Σ' is $\mathcal{H}_{\mathcal{L}'}$–consistent by Proposition 3.11. ■

4.2 Building the Structures

For a first–order language $\mathcal{L}$, we let $CT\,\mathcal{L}$ denote the set of all *closed* $\mathcal{L}$–terms, i.e. terms without any variables. Of course, $CT\,\mathcal{L} = \emptyset$ if $\mathcal{L}$ has no constant symbols.

Let Σ be a set of $\mathcal{L}$–formulae. Using Σ, we now impose on $CT\,\mathcal{L}$ the concepts necessary to turn it into an $\mathcal{L}$–structure, which we will call $CT\,\mathcal{L}_\Sigma$. As the universe of $CT\,\mathcal{L}_\Sigma$, we choose $CT\,\mathcal{L}$. The interpretations of function and constant symbols are straight forward and do not rely on Σ:

For $t_1, \ldots, t_{\mu(j)} \in CT\,\mathcal{L}$ $(j \in J)$, set

$$f_j^{CT\,\mathcal{L}_\Sigma}(t_1, \ldots, t_{\mu(j)}) := f_j(t_1, \ldots, t_{\mu(j)}),$$

and for $k \in K$,

$$c_k^{CT\,\mathcal{L}_\Sigma} := c_k.$$

The interpretation of relation symbols is a bit more tricky, since we are now introducing Σ.

For $t_1, \ldots, t_{\lambda(i)} \in CT\,\mathcal{L}$ $(i \in I)$, set

$$R_i^{CT\,\mathcal{L}_\Sigma}(t_1, \ldots, t_{\lambda(i)}) \text{ iff } \Sigma \vdash R_i(t_1, \ldots, t_{\lambda(i)}).$$

Note that the above construction could have been done on the set of *all* $\mathcal{L}$–terms instead of restricting it to $CT\,\mathcal{L}$. We would then have found the resulting Henkin–structure (cf. Definition 4.6) as a substructure (cf. Definition 2.19) of $Trm\,\mathcal{L}$, provided there are any constant symbols in $\mathcal{L}$. If there are none, $CT\,\mathcal{L}$ is empty and therefore *per definitionem* not an $\mathcal{L}$–structure.

Definition 4.4. Let $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ be a Hilbert–system and $\Sigma \subseteq Sen\,\mathcal{L}$. By $\sim_{\mathcal{H},\Sigma}$ we denote the following equivalence–relation on the set of $\mathcal{L}$–terms:

$$t_1 \sim_{\mathcal{H},\Sigma} t_2 \qquad \text{iff} \qquad \Sigma \vdash_{\mathcal{H}} t_1 \doteq t_2.$$

As usually for equivalence–relations, we denote by $[t]_{\sim_{\mathcal{H},\Sigma}}$ the $\sim_{\mathcal{H},\Sigma}$–class of t, i.e. the set of all $\mathcal{L}$–terms t' with $t' \sim_{\mathcal{H},\Sigma} t$.

It is easy to see that, if $\mathcal{H}_1 = \langle \mathcal{L}, \mathcal{R}_1 \rangle$ and $\mathcal{H}_2 = \langle \mathcal{L}, \mathcal{R}_2 \rangle$ are adequate Hilbert–systems over the same language $\mathcal{L}$, $\sim_{\mathcal{H}_1,\Sigma} = \sim_{\mathcal{H}_2,\Sigma} = \sim_{\mathcal{H}_\mathcal{L},\Sigma}$ for any $\Sigma \subseteq Sen\,\mathcal{L}$. Therefore, in the context of adequate Hilbert–systems, we write $\sim_\Sigma$ instead of $\sim_{\mathcal{H},\Sigma}$.

Proposition 4.5. *Let* $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ *be an adequate Hilbert–system, and assume that* $CT\,\mathcal{L}_\Sigma \neq \emptyset$*. Then for any* $\Sigma \subseteq Sen\,\mathcal{L}$*,* $\sim_\Sigma$ *is an* $\mathcal{L}$*–congruence on* $CT\,\mathcal{L}_\Sigma$*.*

From this result and Proposition 2.18, it follows immediately that the following definition is valid.

Definition 4.6. Let $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ be an adequate Hilbert–system, and let $\Sigma \subseteq Sen\,\mathcal{L}$. Assume that $\mathcal{L}$ has at least one constant symbol. Then the $\mathcal{L}$–structure $CT\,\mathcal{L}_\Sigma/\sim_\Sigma$ is called the *Henkin–structure* for $\mathcal{L}$ modulo Σ.

To simplify notation, we write $CT\,\mathcal{L}/\Sigma$ instead of $CT\,\mathcal{L}_\Sigma/\sim_\Sigma$. It is worth noticing that

$$card(|\,CT\,\mathcal{L}/\Sigma|) \leq card(CT\,\mathcal{L}) \leq card\,\mathcal{L}$$

for any first–order language $\mathcal{L}$.

Proposition 4.7. *Let* $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ *be an adequate Hilbert–system, and let* $\Sigma \subseteq Sen\,\mathcal{L}$*. Assume that*

(i) Σ *is maximally* $\mathcal{H}$*–consistent, and*

(ii) Σ *has witnesses.*

Then, for all $\alpha \in Sen\,\mathcal{L}$*,*

$$\alpha \in \Sigma \qquad \textit{iff} \qquad CT\,\mathcal{L}/\sim_\Sigma \models \alpha.$$

Especially, $CT\,\mathcal{L}/\sim_\Sigma$ *is a model for* Σ*.*

5 Completeness, Correctness, Compactness

The properties mentioned in the title of this section state that a system of formal logic is well-behaved with respect to its models, and that there is a close connection between deductions and validity.

5.1 Correctness

Correctness expresses the fact that everything that can be deduced formally may also be proven to be correct by meta–mathematical considerations.

Definition 5.1. Let $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ be a Hilbert–system for the formal language $\mathcal{L}$. Then $\mathcal{H}$ is said to be *correct* if, for all $\Sigma \subseteq Fml\,\mathcal{L}, \varphi \in Fml\,\mathcal{L}$,

$$\Sigma \vdash_{\mathcal{H}} \varphi \qquad \text{implies} \qquad \Sigma \Vdash \varphi.$$

Theorem 5.2. *$\mathcal{H}_{\mathcal{L}}$ is correct for any formal language $\mathcal{L}$.*

Sketch of Proof: It suffices to show that for all $\langle \Sigma, \varphi \rangle \in \mathcal{R}_{\mathcal{L}}$, $\Sigma \Vdash \varphi$. The rest follows by induction over the length of deductions. ■

Corollary 5.3. *Every adequate Hilbert–system is correct.*

Corollary 5.4. *Let $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ be an adequate Hilbert–system, and let $\Sigma \subseteq Sen\,\mathcal{L}$. Then:*

If Σ has a model, then Σ is $\mathcal{H}$–consistent.

Sketch of Proof: Assume $\mathcal{A} \models \Sigma$. If Σ were $\mathcal{H}$–inconsistent, then $\Sigma \vdash \alpha \wedge \neg\alpha$ for some $\alpha \in Sen\,\mathcal{L}$. By correctness, $\mathcal{A} \models \alpha \wedge \neg\alpha$, which is clearly impossible. ■

5.2 Completeness

Simply put, a formal system is complete if its system of formal deductions is large enough to include all deductions made for structures in metamathematical contexts.

Definition 5.5. Let $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ be a Hilbert–system for the formal language $\mathcal{L}$. Then $\mathcal{H}$ is said to be *complete* if for all $\Sigma \subseteq Sen\,\mathcal{L}, \varphi \in Sen\,\mathcal{L}$,

$$\Sigma \Vdash \varphi \qquad \text{implies} \qquad \Sigma \vdash_{\mathcal{H}} \varphi.$$

Theorem 5.6. *Let $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ be an adequate Hilbert–system. Then for any $\Sigma \subseteq Sen\,\mathcal{L}$:*

If Σ is $\mathcal{H}$–consistent, then Σ has a model.

Sketch of Proof: Assume Σ is $\mathcal{H}_{\mathcal{L}}$–consistent. By Proposition 4.3, there is an extension $\mathcal{L}'$ of $\mathcal{L}$ and a consistent $\Sigma' \subseteq Sen\,\mathcal{L}'$ with $\Sigma \subseteq \Sigma'$ such that Σ' has witnesses and is maximally consistent. By Proposition 4.7, $CT\,\mathcal{L}' / \sim_{\Sigma'}$ is a model of Σ' and therefore also of Σ. ∎

Theorem 5.7. *Adequate Hilbert–systems are complete.*

Sketch of Proof: Let $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ be an adequate Hilbert–system. If $\Sigma \nvdash_{\mathcal{H}} \varphi$, then $\Sigma \cup \{\varphi\}$ is $\mathcal{H}$–consistent and therefore has a model $\mathcal{A}$. Then $\mathcal{A} \models \Sigma$, but $\mathcal{A} \not\models \varphi$, hence $\Sigma \nVdash \varphi$. ∎

5.3 Strong Completeness, Compactness

Summing up the results of the last section:

Theorem 5.8 (Strong Completeness). *Let $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ be an adequate Hilbert–system. Then for $\Sigma \subseteq Sen\,\mathcal{L}$,*

Σ has a model iff Σ is $\mathcal{H}$–consistent.

Theorem 5.9 (Compactness–Theorem). *Let $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ be an adequate Hilbert–system. Then for $\Sigma \subseteq Sen\,\mathcal{L}$,*

Σ has a model iff every finite subset of Σ has a model.

5.4 The Downward Löwenheim–Skolem Property

The Downward Löwenheim–Skolem Theorem for first–order logic states that it is always possible to find small models for consistent sets of sentences, i.e. models whose cardinality is not larger than the cardinality of the underlying language.

Theorem 5.10. *Let $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ be an adequate Hilbert–system. Then for $\Sigma \subseteq Sen\,\mathcal{L}$,*

Σ has a model iff Σ has a model with cardinality at most $card\,\mathcal{L}$.

Sketch of Proof: This is nothing more than a corollary of the proof of the Completeness Theorem: The model $CT\,\mathcal{L}'/\Sigma' \models \Sigma'$ clearly has the desired cardinality. ■

In the original formulation, only countable languages were considered, therefore the Theorem's original statement was

> Every consistent set of sentences has a countable or finite model.

5.5 The Upward Löwenheim–Skolem Property

Looking up instead of down, there is no boundary to the size of models beyond the cardinality of the underlying language.

Theorem 5.11. *Let $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ be an adequate Hilbert–system, $\Sigma \subseteq Sen\,\mathcal{L}$, and let $\kappa \geq card\,\mathcal{L}$ be an infinite cardinal.*

If Σ has a model of cardinality κ, then Σ has a model of cardinality β for all $\beta \geq \kappa$.

Sketch of Proof: Let $\mathcal{L}'$ be a formal language resulting from adding to $\mathcal{L}$ a set $\{c_\gamma\,;\, \gamma < \beta\}$ of new constant symbols. Let $\Sigma' \subseteq Sen\,\mathcal{L}'$ be given by

$$\Sigma' = \Sigma \cup \{\neg c_{\gamma_1} \doteq c_{\gamma_2}\,;\, \gamma_1 < \gamma_2 < \beta\}.$$

Then by the Compactness Theorem, Σ' has a model, which must be of cardinality β. Moreover, this model of Σ' is also a model of Σ. ■

Note that the constraints on κ are necessary: there are sets of sentences with only finite models.

6 Stronger Versions

6.1 The Downward Löwenheim–Skolem Property Revisited

The proof of the result in this subsection lies beyond the scope of these notes. See [chk90] for details.

Theorem 6.1. *Let $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ be an adequate Hilbert–system, $\Sigma \subseteq \mathit{Sen}\,\mathcal{L}$, and let $\mathcal{B}$ be a model of Σ with $\mathit{card}\,\mathcal{B} \geq \mathit{card}\,\mathcal{L}$. Then there is an model $\mathcal{A}$ of Σ such that*

(i) $\mathit{card}\,\mathcal{A} \leq \mathit{card}\,\mathcal{L}$, and

(ii) $\mathcal{A} \preceq_{\mathcal{L}} \mathcal{B}$.

Especially, $\mathcal{A} \equiv_{\mathcal{L}} \mathcal{B}$.

This version of the Downward Löwenheim–Skolem Theorem is stronger than Theorem 5.10, since it states that the elements of the "small" model have the same first–order properties within the "small" model as they have in the "large" model. The proof requires the notion of the *elementary diagram.*

6.2 The Upward Löwenheim–Skolem Property Revisited

The proof of the result in this subsection lies beyond the scope of these notes. See [chk90] for details.

Theorem 6.2. *Let $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ be an adequate Hilbert–system, $\Sigma \subseteq \mathit{Sen}\,\mathcal{L}$, and let $\mathcal{A}$ be a model of Σ with $\mathit{card}\,\mathcal{A} \geq \mathit{card}\,\mathcal{L}$. Then:*

For all $\kappa \geq \mathit{card}\,\mathcal{A}$ there is a model $\mathcal{B}$ of Σ with $\mathit{card}\,\mathcal{B} = \kappa$ and $\mathcal{A} \preceq_{\mathcal{L}} \mathcal{B}$.

Especially, $\mathcal{A} \equiv_{\mathcal{L}} \mathcal{B}$.

6.3 Vaught's Test

The Vaught–Test is a corollary to the Löwenheim–Skolem Theorems in their strong forms (i.e. Theorems 6.1 and 6.2), and it provides a test for complete theories.

Theorem 6.3. *Let $\mathcal{H} = \langle \mathcal{L}, \mathcal{R} \rangle$ be an adequate Hilbert–system and $\Sigma \subseteq Sen\,\mathcal{L}$. Assume that for some infinite cardinal $\kappa \geq card\,\mathcal{L}$, any two models of Σ with cardinality κ are $\mathcal{L}$–isomorphic. Then Σ is $\mathcal{L}$–complete.*

Sketch of Proof: Take any two models $\mathcal{A}$ and $\mathcal{B}$ of Σ. By the Löwenheim–Skolem Theorems, we find $\mathcal{A}' \equiv_{\mathcal{L}} \mathcal{A}$ and $\mathcal{B}' \equiv_{\mathcal{L}} \mathcal{B}$ with $card\,\mathcal{A}' = card\,\mathcal{B}' = \kappa$. Since $\mathcal{A}' \cong_{\mathcal{L}} \mathcal{B}'$ by assumption, we get

$$\mathcal{A} \equiv_{\mathcal{L}} \mathcal{A}' \cong_{\mathcal{L}} \mathcal{B}' \equiv_{\mathcal{L}} \mathcal{B},$$

hence $\mathcal{A} \equiv_{\mathcal{L}} \mathcal{B}$. ■

7 Undecidability

First Order Logic is — from the viewpoint of expressive power — quite powerful. This is one of the conclusions we can draw from the Completeness Theorem 5.6. The price we pay for formal systems with great expressive power is that algorithmic procedures are not enough to decide provability.

In this section, in the context of a formal language $\mathcal{L}$, $\vdash$ stands for deducibility in $\mathcal{H}_\mathcal{L}$ and therefore in any adequate Hilbert–system: $\quad \vdash = \vdash_{\mathcal{H}_\mathcal{L}}$.

Theorem 7.1. *There are a formal language $\mathcal{L}$ and $\Sigma \subseteq Sen\,\mathcal{L}$ such that the set $\{\alpha \in Sen\,\mathcal{L}\,;\,\Sigma \vdash \alpha\}$ is not recursive.*

It is understood that *recursive* means *recursive under some enumeration*. To be just a little more precise, suppose we are given an enumeration of $Sen\,\mathcal{L}$, i.e. an injective function $e : Sen\,\mathcal{L} \longrightarrow \mathbb{N}$. Then $\{e(\alpha)\,;\,\Sigma \vdash \alpha\}$ is not a recursive set.

There are several examples of appropriate formal languages and Σ. One of the simplest examples is the following.

Example 7.2 (Combinatory Logic). Let $\mathcal{L}_{CL}$, the formal language of *Combinatory Logic*, contain two constant symbols, denoted by S and K, and a binary function symbol, denoted by $\cdot$ (using infix–notation). Let

$$\Sigma_{CL} := \{\forall x \forall y (K \cdot x) \cdot y \doteq x, \forall x \forall y \forall z ((S \cdot x) \cdot y) \cdot z \doteq (x \cdot z) \cdot (y \cdot z), \neg S \doteq K\}.$$

Then by results of Curry, Church and Rosser, there exists no algorithm to decide uniformly, for any two terms t_1 and t_2, whether or not $\Sigma \vdash t_1 \doteq t_2$. The main reason for this undecidability lies in the fact that the recursive functions are representable in the theory axiomatized by Σ in the following sense: There are distinct $\mathcal{L}_{CL}$–terms $N_0, N_1, \ldots$ (so–called *numerals*) such that for all recursive $F : \mathbb{N}^k \longrightarrow \mathbb{N}$, there is a $\mathcal{L}_{CL}$–term T_F such that

$$\Sigma_{CL} \vdash (\ldots (T_F \cdot N_{x_1}) \ldots) \cdots N_{x_k} \doteq N_{F(x_1,\ldots,x_k)}$$

for all $x_1, \ldots, x_k \in \mathbb{N}$.

From this and so–called *fixed–point properties*, the undecidability follows from an argument similar to that found in proofs of the unsolvability of the *Halteproblem* for Turing-machines.

For details, see [bar84] or [spr04].

An alternative way of proving the undecidability of first–order theories in general uses *Post's Correspondence Problem.*

Definition 7.3. Let $\{0,1\}^*$ denote the set of strings (i.e. finite sequences) of 0's and 1's. A *correspondence problem* is a finite sequence of pairs

$$K = \langle\langle x_1, y_1\rangle, \ldots, \langle x_n, y_n\rangle\rangle$$

where $x_k, y_k \in \{0,1\}^*$, $n \in \mathbb{N}$. A *solution* for the correspondence problem K is a finite sequence $\langle i_1, \ldots, i_m\rangle$ with $i_k \in \{1, \ldots, n\}$ such that the concatenations $x_{i_1} \cdots x_{i_m}$ and $y_{i_1} \cdots y_{i_m}$ are identical (as strings over $\{0,1\}$).

Theorem 7.4. *There is no algorithm which decides uniformly whether a given correspondence problem has a solution or not.*

For the proof, readers are referred to [her78].

Theorem 7.5. *Let $\mathcal{H} = \langle\mathcal{L}, \mathcal{R}\rangle$ be an adequate Hilbert–system where $\mathcal{L}$ contains a binary relation symbol P, two unary function symbols f_0, f_1 and a constant symbol a. Then there is no algorithm which for any $\varphi \in Fml\mathcal{L}$ answers "yes" if $\vdash_{\mathcal{H}} \varphi$, and "no" otherwise.*

Sketch of Proof: For $j_1 j_2 \ldots j_s \in \{0,1\}^*$ and $t \in Trm\,\mathcal{L}$, define $f_{j_1 j_2 \ldots j_s}(t) \in Trm\,\mathcal{L}$ by

$$f_{j_1 j_2 \ldots j_s}(t) := f_{j_s}(f_{j_{s-1}} \ldots (f_{j_2}(f_{j_1}(t))) \ldots).$$

For $K = \langle\langle x_1, y_1\rangle, \ldots, \langle x_n, y_n\rangle\rangle$ a correspondence problem, define φ_K by

$$\varphi_K := \rho \wedge \sigma \rightarrow \tau,$$

where

$$\begin{aligned}
\rho &:= P(f_{x_1}(a), f_{y_1}(a)) \wedge \ldots \wedge P(f_{x_n}(a), f_{y_n}(a)),\\
\sigma &:= \forall u \forall v (P(u,v) \rightarrow (P(f_{x_1}(u), f_{y_1}(v)) \wedge \ldots \wedge P(f_{x_n}(u), f_{y_n}(v)))),\\
\tau &:= \exists z P(z,z).
\end{aligned}$$

Then the equivalence

$$\varphi_K \text{ is valid} \qquad \text{iff} \qquad k \text{ has a solution}$$

can be seen as follows:

$\Rightarrow$ Define $\mathcal{A}$ to be the $\mathcal{L}$–structure $\mathcal{A} = \{\{0,1\}^*, P^{\mathcal{A}}, f_0^{\mathcal{A}}, f_1^{\mathcal{A}}, \bot\}$ where $P^{\mathcal{A}}(x,y)$ iff there is a sequence $\langle i_1, \ldots, i_s \rangle$ of indices from $\{1, \ldots, n\}$ such that $x = x_{i_s} \cdots x_{i_s}$ and $y = y_{i_s} \cdots y_{i_s}$, $f_0^{\mathcal{A}}(x) := x0$ (concatenation), $f_1^{\mathcal{A}}(x) := x1$ and $\bot$ denotes the empty sequence. Then $\mathcal{A} \models \rho \wedge \sigma$, and since φ_K is assumed to be valid, $\mathcal{A} \models \tau$, which is easily seen to state that K has a solution.

$\Leftarrow$ Assume K has a solution. Let $\mathcal{A}$ be any $\mathcal{L}$–structure, and define $\eta : \{0,1\}^* \longrightarrow |\mathcal{A}|$ inductively by $\eta(\bot) := a^{\mathcal{A}}$, $\eta(x0) := f_0^{\mathcal{A}}(\eta(x))$ and $\eta(x1) := f_1^{\mathcal{A}}(\eta(x))$. Without loss of generality, we may assume $\mathcal{A} \models \rho \wedge \sigma$, since otherwise trivially $\mathcal{A} \models \varphi_K$. But then, for a solution $x_{i_1} \cdots x_{i_s} = y_{i_1} \cdots y_{i_s} =: d$ for K, we can see that $\mathcal{A} \models P^{\mathcal{A}}(\eta(d), \eta(d))$, hence $\mathcal{A} \models \tau$ and finally $\mathcal{A} \models \varphi_K$.

■

8 Model Theory

In this section, we give a brief outline of the model-theoretic results for first–order logic.

8.1 Model Classes and Theories

We take a dual approach of Section 3.5.

For a first–order language $\mathcal{L}$, we let $Str\,\mathcal{L}$ denote the class of $\mathcal{L}$–structures.

Definition 8.1. Let $\mathcal{L}$ be a first–order language and $\mathbf{K} \subseteq Str\,\mathcal{L}$.

(i) A class $\mathbf{K} \subseteq Str\,\mathcal{L}$ is called $\mathcal{L}$*–elementary* or $\mathcal{L}$*–axiomatizable* if $\mathbf{K} = \{\mathcal{A} \in Str\,\mathcal{L}\,;\ \mathcal{A} \models \Sigma\}$ for some $\Sigma \subseteq Sen\,\mathcal{L}$.

(ii) If in (i) Σ can be chosen to be finite, then $\mathbf{K}$ is called *finitely* $\mathcal{L}$*–axiomatizable.*

(iii) If in (i) Σ can be chosen to be a singleton set, then $\mathbf{K}$ is called $\mathcal{L}$*–basic–elementary.*

The implications

$$\text{basic–elementary} \Longrightarrow \text{finitely axiomatizable} \Longrightarrow \text{elementary}$$

are obvious, as are the following facts.

Proposition 8.2. *Let $\mathcal{L}$ be a first–order language. Then*

(i) $\mathbf{K} \subseteq Str\,\mathcal{L}$ *is* $\mathcal{L}$*–basic–elementary iff it is finitely* $\mathcal{L}$*–axiomatizable.*

(ii) $\mathbf{K}$ *is* $\mathcal{L}$*–basic–elementary iff the complementary class* $Str\,\mathcal{L} \smallsetminus \mathbf{K}$ *is* $\mathcal{L}$*–basic–elementary.*

(iii) Every $\mathcal{L}$*–elementary class is the intersection of* $\mathcal{L}$*–basic–elementary classes.*

Sketch of Proof:

(i) $\mathcal{A} \models \{\alpha_1, \ldots, \alpha_n\}$ iff $\mathcal{A} \models \alpha_1 \wedge \ldots \wedge \alpha_n$.

(ii) $\mathbf{K} = \{\mathcal{A} \in Str\,\mathcal{L}\,;\ \mathcal{A} \models \alpha\}$ iff $Str\,\mathcal{L} \smallsetminus \mathbf{K} = \{\mathcal{A} \in Str\,\mathcal{L}\,;\ \mathcal{A} \models \neg\alpha\}$.

(iii) $\{\mathcal{A} \in Str\,\mathcal{L}\,;\ \mathcal{A} \models \Sigma\} = \bigcap_{\alpha\in\Sigma}\{\mathcal{A} \in Str\,\mathcal{L}\,;\ \mathcal{A} \models \alpha\}$.

■

Recall Definition 2.22 for the notion of elementary equivalence.

Proposition 8.3. *$\mathcal{L}$–elementary classes are closed under $\equiv_{\mathcal{L}}$.*

Note that the converse does not hold, i.e. there are classes of $\mathcal{L}$–structures which are closed under $\equiv_{\mathcal{L}}$ but which are not $\mathcal{L}$–elementary. See Theorem 8.10.

8.2 Algebraic Aspects of Elementary Classes

There are ways of characterizing elementary classes without the use of syntactic notions.

Definition 8.4 (Direct Product). If $\langle \mathcal{A}_s\,;\ s \in S\rangle$, $S \neq \emptyset$, is a family of $\mathcal{L}$–structures, then the *direct product* $\prod_{s\in S}\mathcal{A}_s$ is the $\mathcal{L}$–structure $\mathcal{B}$ given by

- $|\mathcal{B}| = \prod_{s\in S}|\mathcal{A}_s|$;
- $f_j^{\mathcal{B}}(b_1,\ldots,b_{\mu(j)}) = \langle f_j^{\mathcal{A}_s}((b_1)_s,\ldots,(b_{\mu(j)})_s)\,;\ s\in S\rangle$;
- $R_i^{\mathcal{B}}(b_1,\ldots,b_{\lambda(i)})$ iff $R_i^{\mathcal{A}_s}((b_1)_s,\ldots,(b_{\lambda(i)})_s)$ for all $s \in S$;
- $c_k^{\mathcal{B}} = \langle c_k^{\mathcal{A}_s}\,;\ s\in S\rangle$.

Definition 8.5 (Ultrafilter). Let $S \neq \emptyset$ be any set. Then a system $\mathcal{U}$ of subsets of S is an *ultrafilter* over S if

(i) $\emptyset \notin \mathcal{U} \neq \emptyset$;

(ii) $U, V \in \mathcal{U}$ implies $U \cap V \in \mathcal{U}$;

(iii) $U \in \mathcal{U}$, $U \subseteq V \subseteq S$ implies $V \in \mathcal{U}$;

(iv) $U \notin \mathcal{U}$ implies $S \smallsetminus U \in \mathcal{U}$;

for all $U, V \subseteq S$.

Definition 8.6. Let $\langle \mathcal{A}_s \,;\, s \in S \rangle$, $S \neq \emptyset$, be a family of $\mathcal{L}$–structures, $\mathcal{B} = \prod_{s \in S} \mathcal{A}_s$ the direct product of $\langle \mathcal{A}_s \,;\, s \in S \rangle$. Let $\mathcal{U}$ be an ultrafilter over S. Then $\sim_{\mathcal{U}}$ denotes the binary relation on $|\mathcal{B}|$ given by

$$b_1 \sim_{\mathcal{U}} b_2 \qquad \text{iff} \qquad \{s \in S \,;\, (b_1)_s = (b_2)_s\} \in \mathcal{U}.$$

Proposition 8.7. *$\sim_{\mathcal{U}}$ as in Definition 8.6 is a congruence on $\prod_{s \in S} \mathcal{A}_s$.*

This is a justification for the following definition.

Definition 8.8 (Ultraproduct). Let $\langle \mathcal{A}_s \,;\, s \in S \rangle$, $S \neq \emptyset$, be a family of $\mathcal{L}$–structures, $\mathcal{B} = \prod_{s \in S} \mathcal{A}_s$ the direct product of $\langle \mathcal{A}_s \,;\, s \in S \rangle$. Let $\mathcal{U}$ be an ultrafilter over S. Then the *ultraproduct* of $\langle \mathcal{A}_s \,;\, s \in S \rangle$ modulo $\mathcal{U}$ is the $\mathcal{L}$–structure $\prod_{s \in S} \mathcal{A}_s / \sim_{\mathcal{U}}$. For reasons of convenience, the notation $\prod_{s \in S} \mathcal{A}_s / \mathcal{U}$ is commonly used.

If for some $\mathcal{L}$–structure $\mathcal{A}$, $\mathcal{A}_s = \mathcal{A}$ for all $s \in S$, then $\prod_{s \in S} \mathcal{A}_s / \sim_{\mathcal{U}}$ is called an *ultrapower*, notation $\mathcal{A}^S / \mathcal{U}$.

Theorem 8.9 (Łoś). *Let $\mathcal{L}$ be a formal language. Let $\prod_{s \in S} \mathcal{A}_s / \mathcal{U}$ be the ultraproduct of $\langle \mathcal{A}_s \,;\, s \in S \rangle$ modulo the ultrafilter $\mathcal{U}$ over S. Then for all $\alpha \in Sen\,\mathcal{L}$,*

$$\prod_{s \in S} \mathcal{A}_s / \mathcal{U} \models \alpha \qquad \textit{iff} \qquad \{s \in S \,;\, \mathcal{A}_s \models \alpha\} \in \mathcal{U}.$$

Especially for Ultrapowers, it follows

$$\mathcal{A}^S / \mathcal{U} \models \alpha \qquad \textit{iff} \qquad \mathcal{A} \models \alpha.$$

Theorem 8.10. *Let $\mathcal{L}$ be a formal language. Then a class $\mathbf{K}$ of $\mathcal{L}$–structures is $\mathcal{L}$–elementary iff*

(i) *$\mathbf{K}$ is closed under $\equiv_{\mathcal{L}}$ and*

(ii) *$\mathbf{K}$ is closed under ultraproducts.*

A purely semantical characterization, avoiding the syntactical notion of elementary equivalence, is given in the following Theorem.

Theorem 8.11. *Let $\mathcal{L}$ be a formal language. Then a class $\mathbf{K}$ of $\mathcal{L}$–structures is $\mathcal{L}$–elementary iff* *(i)* $\mathbf{K}$ *is closed under* $\cong_{\mathcal{L}}$*,*

(ii) $\mathbf{K}$ *is closed under ultraproducts and*

(ii) $Str\,\mathcal{L} \setminus \mathbf{K}$ *is closed under ultrapowers.*

Corollary 8.12. *Let* $\mathcal{L}$ *be a formal language. Then a class* $\mathbf{K}$ *of* $\mathcal{L}$*–structures is* $\mathcal{L}$*–basic–elementary iff both* $\mathbf{K}$ *and* $Str\,\mathcal{L} \setminus \mathbf{K}$ *are closed under* $\cong_{\mathcal{L}}$ *and ultrapowers.*

8.3 Compactness and Löwenheim–Skolem Revisited

Using elementary and basic–elementary classes, the Compactness Theorem 5.9 may be reformulated as follows.

Theorem 8.13. *If, for* $i \in I$*,* $\mathbf{K}_i$ *are basic–elementary classes such that the intersection class of finitely many of the* $\mathbf{K}_i$*'s is always non–empty, then* $\bigcap_{i\in I} \mathbf{K}_i \neq \emptyset$*.*

Another way of reformulation uses ultraproducts and provides a way of "contructing" a model for Σ from the models of the finite subsets of Σ.

Theorem 8.14. *Assume that for* $\Sigma \subseteq Sen\,\mathcal{L}$*, every finite* $\Theta \subseteq \Sigma$ *has a model* $\mathcal{A}_\Theta$*. Then there is an ultraproduct of* $\langle \mathcal{A}_\Theta\,;\,\Theta \subseteq \Sigma$ *finite*$\rangle$ *which is a model for* Σ*.*

Sketch of Proof: Set $S := \{\Theta \subseteq \Sigma\,;\,\Theta \text{ finite}\}$, and for $\Theta \in S$ let $\mathcal{A}_\Theta$ be a model for Θ. Then there is an ultrafilter $\mathcal{U}$ over S such that for all $\alpha \in \Sigma$, $T_\alpha := \{\Theta \in S\,;\,\varphi \in \Theta\} \in \mathcal{U}$. From this it follows that $\prod_{\Theta\in S} \mathcal{A}_\Theta/\mathcal{U}$ is a model for Σ. ■

The following is an elegant formulation of one of the consequences of the Downward Löwenheim–Skolem Theorem 5.10.

Theorem 8.15. *In every basic–elementary class there is a countable structure.*

8.4 Non–Standard Analysis

Non–standard analysis is a nice example for the use of ultraproducts. It deals with so–called *non–standard models* of the first–order theory of the real numbers as an ordered algebraic structure.

Suppose $\mathcal{L}_{\mathbb{R}}$ is a formal language appropriate for expressing the algebraic and order–theoretic properties of $\mathbb{R}$. From the Compactness Theorem 5.9 we can conclude that there is a model for the $\mathcal{L}_{\mathbb{R}}$–theory of $\mathbb{R}$ together with the infinite set Σ of $\mathcal{L}$–sentences given by

$$\Sigma := \{c \geq n\ ;\ n \in \mathbb{N}\},$$

where c is a *new* constant symbol. Then, the interpretation of c in this model must be an element which is larger than any natural number, i.e. an *infinite element*. The punch line therefore is:

It is consistent to assume that there are infinitely large numbers.

We now introduce the formal framework for non–standard models, using ultraproducts.

Definition 8.16. Assume $\mathcal{L}$ is a formal language having—possibly among other non–logical symbols—a binary relation symbol, denoted by $\leq$. Then an $\mathcal{L}$–structure $\mathcal{A}$ is an *ordered structure* if the interpretation of $\leq$ in $\mathcal{A}$ is a partial order. An *increasing sequence* in $\mathcal{A}$ is a sequence $\langle a_n\ ;\ n \in \mathbb{N}\rangle$ such that $a_i \leq^{\mathcal{A}} a_{i+1}$ for all $n \in \mathbb{N}$. An increasing sequence in $\mathcal{A}$ is called *co–final* in $\mathcal{A}$ if for all $a \in |\mathcal{A}|$, there is an $i \in \mathbb{N}$ such that $a \leq^{\mathcal{A}} a_i$. If there is a co–final sequence in $\mathcal{A}$, then $\mathcal{A}$ is called *(upward) unbounded*.

By Theorem 8.9, any ultrapower of a structure is elementary equivalent to the original structure. Still, we now define an ultraproduct on an arbitrary ordered structure whose elements are — so to speak — infinitely large with respect to the original elements. These arguments indicate that the notion of "being infinitely large" is not expressible in a first–order language.

Proposition 8.17. *Let S be an infinite set. Then, there is an ultrafilter $\mathcal{U}$ over S such that for all $U \subseteq S$,*

$$\text{if } S \setminus U \text{ is finite, then } U \in \mathcal{U}.$$

Sketch of Proof: Call $U \subseteq S$ *co–finite* iff $S \setminus U$ is finite. The set of all co–finite subsets of S is a so–called *filter*, i.e. it is closed under binary intersection and supersets, and it is not empty nor equal to the whole of S. Using Zorn's Lemma, it is straight forward to show that any filter is contained in a maximal filter, and ultrafilters are exactly the maximal filters. ■

Definition 8.18. Let $\mathcal{A}$ be a structure. Let S be an infinite set, $\mathcal{U}$ be an ultrafilter over S as in Proposition 8.17. Then the ultrapower $\mathcal{A}^S/\mathcal{U}$ is called *a non–standard model* of the theory of $\mathcal{A}$.

Proposition 8.19. *Let $\mathcal{A}$ be an unbounded ordered $\mathcal{L}$–structure with partial order $\leq^{\mathcal{A}}$, and let $\mathcal{B} = \mathcal{A}^{\mathbb{N}}/\mathcal{U}$ be a non–standard model of the theory of $\mathcal{A}$. Then:*

(i) The mapping $a \mapsto [\langle a\,;\, s \in S\rangle]_{\mathcal{U}} =: \overline{a}$ defines an $\mathcal{L}$–isomorphism from $\mathcal{A}$ onto an elementary substructure of $\mathcal{B}$.

(ii) There are $b \in |\mathcal{B}|$ such that $\overline{a} \leq^{\mathcal{B}} b$ for all $a \in |\mathcal{A}|$.

Sketch of Proof: The proof of (i) requires tools outside the scope of these notes and is therefore omitted. See [chk90] for details.

For (ii), take any co–final sequence $\langle a_n\,;\, n \in \mathbb{N}\rangle$ in $\mathcal{A}$ and set $b := [\langle a_n\,;\, n \in \mathbb{N}\rangle]_{\mathcal{U}}$. Then for any $a \in |\mathcal{A}|$, $\{i \in \mathbb{N}\,;\, a_i \leq^{\mathcal{A}} a\}$ is finite (by the definition of "co–final") and thus not in $\mathcal{U}$, so its complement is in $\mathcal{U}$, from which $\overline{a} \leq^{\mathcal{B}} b$ follows easily. ■

Setting $\mathcal{A} = \mathbb{R}$, we get the field of mathematical research called *non–standard analysis*. The inifinite elements of the non–standard structure obey the algebraic laws of $\mathbb{R}$. In particular, they have multiplicative inverses since they are all different from 0. These elements $1/b$ for infinite b are *infinitesimally small* in the sense that they are closer to 0 than any real number (isomorphically embedded into the non–standard model), but still they are non–zero.

In non–standard models of $\mathbb{R}$, we loose an important property which is valid in the ordered field $\mathbb{R}$: *The Archimedean Property*:

Let $\mathbb{F}$ be an ordered field. Then $\mathbb{F}$ has the *Archimedean Property* (AP) iff

the additive subgroup generated by 1 is co–final in $\mathbb{R}$.

The idea behind it is better formulated as

if the number of summands is chosen large enough,
any real number lies below a (finite!) sum of 1s .

Assume now that $\mathcal{L}_{OF}$ is the first–order language for ordered fields, i.e. $\mathcal{L}_{OF} = \{+, \times, -, 0, 1, \leq\}$. Let $\Theta_{\mathbb{R}}$ be the (complete!) $\mathcal{L}_{OF}$–theory of $\mathbb{R}$, and let $\mathbb{R}^*$ be a non–standard model of $\Theta_{\mathbb{R}}$. From the Theorem of Łoś 8.9, we know that $\mathbb{R}^* \equiv_{\mathcal{L}_{OF}} \mathbb{R}$. Considering $s := [\langle 0, 1, 2, \ldots \rangle] \in |\mathbb{R}^*|$, we see that there is no sum of $\overline{1}$s larger than s. Therefore, (AP) is not in the $\mathcal{L}_{OF}$–theory of $\mathbb{R}$, i.e. (AP) is not expressible in $\mathcal{L}_{OF}$.

If we had chosen to formulate (AP) (wrongly!) as "any real number lies below a natural number", we could have run into trouble. One could enrich $\mathcal{L}_{OF}$ to $\mathcal{L}_{OF}^+ := \{+, \times, -, 0, 1, \leq, N\}$ by adding a new unary relation symbol N, interpreted as "is a natural number":

$$N^{\mathbb{R}}(x) \text{ iff } x \in \mathbb{N}.$$

Assuming the erroneous formalization, (AP) becomes expressible in $\mathcal{L}_{OF}^+$ as

$$\forall x \exists y (N(y) \wedge x \leq y \wedge \neg(x \doteq y))$$

and would therefore also hold in non–standard models. But note that for s as above, we would need a *non–standard natural number* for y, i.e. an element in the extension of N in the ultraproduct $\mathbb{R}^*$, which is not the value of some natural number under the canonical embedding $x \mapsto \overline{x} \in$ from $\mathbb{R}$ into $\mathbb{R}^*$.

9 Abstract Model Theory

In this section, we give a brief motivation for the field of research called *abstract model theory*. The main difference between the abstract and the "ordinary" model theory we introduced in the previous sections lies in the fact that abstract model theory explores the correspondence between the notion of satisfaction $\models$ and classes of models by taking a categorical point of view.

What characterizes first–order logic? One of the most obvious observation is the finiteness of the notions involved. Terms, formulae and deductions are all finite constructs. Structures and model–classes, on the other hand, are not. The Compactness Theorem links the hereditary finiteness of the syntactic constructs with the (actual and potential) infiniteness of the semantic side of the coin.

9.1 Towards Lindstrom's Characterization

We are free to expand our notion of deduction in several ways. E.g., we could introduce new kinds of quantifiers which postulate the existence of a set of elements with a specific cardinality, something like "there exist at least uncountably many ...". Or we could break the boundaries of finiteness by allowing for infinite formulae: for $i \in \mathbb{N}$, formulae φ_i, $\bigwedge_{i \in \mathbb{N}} \varphi_i$ would then be a formula.

Investigating such expansions of the underlying language provides insight in the special rôle first–order logic has as a framework for formalizing mathematical concepts.

Lindstrom's result states that basically,

> First–order logic is the only logic which satisfies both the Downward Löwenheim–Skolem Property and the Compactness Property.

References

[bar84] Barendregt, Henk: The lambda calculus, its syntax and semantics. North-Holland, Amsterdam 1984.

[chk90] Chang, C.C., and Keisler, H.J.: Model Theory. North-Holland, Amsterdam 1991.

[her78] Hermes, Hans: Aufzählbarkeit, Entscheidbarkeit, Berechenbarkeit. Springer, Berlin, Heidelberg, New York 1978.

[pre86] Prestel, Alexander: Einführung in die Mathematische Logik und Modelltheorie. Vieweg Studium, Braunschweig 1986.

[spr04] Sprenger, Markus: Kombinatorische Algebra. Unpublished notes. 2004.
http://www.algebraworkbench.net/download/KombinatorischeAlgebra.pdf