Logic required for Set Theory I (see Kunen Ch. I, §§ 13f)

Jakob Kellner, 2011SS

April 22, 2011

1 The case of natural numbers

Assume for simplicity that we add the constant symbol $\emptyset$ for the empty set and the successor function S to our language. I.e., our language is not just $\{\in\}$, but $\{\in, \emptyset, S\}$. This does not change anything since $\emptyset$ and S are definable anyway, see Kunen Ch. I, §13; or Ziegler Skriptum Satz 7.4.

Recall that we can **code** each natural number n as a term $\ulcorner n \urcorner$, defined by induction: $\ulcorner 0 \urcorner := \emptyset$, $\ulcorner 1 \urcorner := S(\emptyset)$, $\ulcorner 1 \urcorner := S(S(\emptyset))$ etc.

We do not claim that $\ulcorner n \urcorner$ is "really" the "true" form of n, or that the coding is in any way natural. The point is that the coding is just a reasonable way to talk about natural numbers in pure set theory without the need to add any additional symbols or non-hederity sets.

Recall that we define (in ZF) ω to be the first limit ordinal (bigger than 0).

Of course the "intention" is to describe the set $\{\ulcorner 0 \urcorner, \ulcorner 1 \urcorner, \dots\}$, but it is important to note that $\omega = \{\ulcorner 0 \urcorner, \ulcorner 1 \urcorner, \dots\}$ is **not** the definition of ω! This "definition" would be an infinite formula, which is not a valid first order formula. And we know that we cannot express $\{\ulcorner 0 \urcorner, \ulcorner 1 \urcorner, \dots\}$ in forst order at all:

Assume (as always) that ZF is consistent. Then we can construct a non-standard model of ZF in the usual way: First extend the language by a new constant symbol c, and extend ZF to a theory T by adding the sentences $c \in \omega$, $c > 0$, $c > 1$, etc. By the compactness theorem, T is consistent, i.e., has a model $\mathcal{M}$. Let $c^{\mathcal{M}}$ be the interpretation of c in $\mathcal{M}$. Then $\mathcal{M} \models c^{\mathcal{M}} \in \omega$, but for all $n \in \mathbb{N}$ we have $\mathcal{M} \models c^{\neq} \ulcorner n \urcorner$.

2 Coding senteces

Completely analogously to natural numbers, we can translate (code) formulas into terms in the language of set theory.

Fix a first-order signature L. For simplicity, assume that L is finite, as it is the case in set theory, where $L = \{\in\}$, or in Peano arithmetic, where L is, e.g., $\{0, 1, +, \cdot, <\}$). (Of course we could just as well use any infinite recursive signature.)

We start with coding the alphabet (i.e., the set of symbols), by "arbitrarily" assigning a natural number n (more exactly the corresponding term $\ulcorner n \urcorner$) to each symbol. E.g., we can assign terms of the form $\ulcorner 3n \urcorner$ to the logical symbols (other than variables): $\ulcorner (\urcorner := \ulcorner 0 \urcorner$, $\ulcorner) \urcorner := \ulcorner 3 \urcorner$, $\ulcorner \wedge \urcorner := \ulcorner 0 \urcorner$, and so on for $\vee$, $\neg$, $\rightarrow$, $\exists$, $\forall$, $=$; we assign $\ulcorner 3n+1 \urcorner$ to the variable symbol v_n, i.e., $\ulcorner v_0 \urcorner := \ulcorner 1 \urcorner$ etc; and we assign terms of the form $\ulcorner 3n+2 \urcorner$ to the logical symbols; e.g., $\ulcorner \in \urcorner := \ulcorner 2 \urcorner$ (of course the logical symbols can be different for other signatures L).

From here on, it is easy (and even "natural") to define all the other notions of mathematical logic inside ZF. (Remark: This is just an instance of the following general claim: All ("normal") mathematical concepts and proofs can be carried out naturally in ZF.)

Syntax:

- We can define (in ZF) "x is a symbol" (which just means that $x = 3 \cdot n + 1$ for some $n \in \omega$ or that x is in a certain finite list of numbers).
- We can define "x is a string", which just means $x \in \omega^{<\omega}$ and $x(l)$ is a symbol for each $l \in \text{dom}(x)$.
- We can define "x is a formula", which is a rather long and tedious but entirely natural definition, e.g., by induction on the length of the string x: $x \in \omega^{n+1}$ is a formula if EITHER there is a Formula $y \in \omega^n$ (this is already defined by induction) such that $x(0) = \ulcorner \neg \urcorner$ and $x(l+1) = y(l)$ for all $0 \leq l < n$, OR etc.
- We can define all the other syntactical properties, such as "v is a variable symbol occuring freely in the formula x", "the formula x is the result of the conjunction of y and z" (or, informally, $x = y \wedge z$), etc etc.

Logical calculus/provability:

- We can define "x is a logical axiom" (again, a tedious case destinction).
- We can define in ZF: "s is a proof in T". (I.e., T is a set of formulas, s is a finite sequence of formulas, each one being a logical axiom or an element of T or follows from previous ones by modus ponens).
- So we can define in ZF the formula "$T \vdash x$" with two free variables T and x that expresses "the formula x is provable in T".
- We can also formulate "The formula x is an instance of the Replacement axiom scheme". (And similarly for separation.)
- We can define the set of all ZF axioms (which we just call ZF). (This is just the smalles subset of $\omega^{<\omega}$ containing all instances of replacement and separation as well as the (finitely many) other axioms.)
- So we can also formulate in ZF the formula ZF$\vdash x$ (here x is the only free variable).

Semantics:

- We can define "$\mathcal{M}$ is an L-structure" (which just says (in the case of $L = \{\in\}$: $\mathcal{M}$ is a pair $\langle M, E \rangle$, M is nonempty, and $E \subseteq M^2$. (If we have other non-logical function or relation symbols we have to modify this accordingly.)
- We can define "$\mathcal{M} \models \varphi(\bar{m})$".
  Actually, later on it will be important to see **how** we can define $\mathcal{M} \models \varphi(\bar{m})$. (We can define it with a very simple formula, which implies that the definition is absolute for transitive models.)
- So we can define the formula "$T \vDash \varphi$" with free variables T and φ that expresses that T semantically implies φ.

Completeness theorem:

- We can now formulate (and prove!) in ZF the completeness theorem ($T \vDash \varphi$ iff $T \vdash \varphi$), as well as the compactness theorem, Skolem Löwenheim (we might need AC for that, though) etc etc.

Again, note that just as in the case of natural numbers, the set of Symbols is **not** defined by $\{\ulcorner \in \urcorner, \ulcorner (\urcorner, \dots, \ulcorner = \urcorner, \ulcorner v_0 \urcorner, \ulcorner v_1 \urcorner, \dots\}$; if a ZF-model $\mathcal{M}$ contains nonstandard natural numbers than it also contain a nonstandard variable-symbol, a formula containing nonstandard symbols, a formula of nonstandard length, a nonstandard proof etc.

A special case of this effect arises in connection with Gödels incompleteness theorem (which will be discussed below): If ZF is consistent, then (by the incompleteness theorem) so is T:=ZF+¬Con(ZF). Let $\mathcal{M}$ be a model of T. Then $\mathcal{M}$ thinks that there is a proof of $\neg 0 = 0$ from ZF; obviously this proof cannot be the code of a real proof but has to be a nonstandard proof.

3 Representing recursive and recursively enumerable sets

Actually, all the previous claims that certain properties are expressible in ZF are special cases of the following important fact:

- A is strongly definable in ZF, if there is a formula φ such that
  - $n \in A$ iff ZF$\vdash \varphi(\ulcorner n \urcorner)$.
  - $n \in \mathbb{N} \setminus A$ iff ZF$\vdash \neg\varphi(\ulcorner n \urcorner)$.
- A is weakly definable in ZF, if there is a formula φ such that
  - $n \in A$ iff ZF$\vdash \varphi(\ulcorner n \urcorner)$.
- $A \subseteq \mathbb{N}$ is recursive, iff A is strongly definable in ZF.

- $A \subseteq \mathbb{N}$ is recursively enumerable, iff A is weakly definable in ZF.

(We do not need ZF here, we get exactly the same result for much weaker theories such as PA or Q.)

The proof is rather straightforward, just describe in ZF what the computer is doing.

4 Fixed Point Lemma, Undefinability of Truth, Incompleteness Theorem

The following is the "Fixed Point Lemma" (14.2 in Kunen): Let $\phi(x)$ be any formula with one free variable x. Then there is a sentence ψ such that

$$\mathrm{ZF} \vdash \psi \leftrightarrow \phi(\ulcorner \psi \urcorner)$$

One important consequence is that truth of a formula is undefinable! We have seen that there is a ZF formula $Pr(x)$ that expresses that x is provable in ZFC; and that there is a ZF formula $M \models x$ (with two free variables M, x) that expresses that x is true in M; but there is no ZF formula $V \models x$ (with one free variable x) that expresses that x is simply true, i.e., holds in the proper class V of all sets. (Given any such formula $V \models \cdot$ There is always a ψ such that $\psi \leftrightarrow \neg V \models (\psi)$ holds, and is even provable in ZF.)

So the important point to remember is: For *sets* M we *can* formulate $M \models x$ inside of ZF; but for proper classes $\mathbf{C}$ we *can not* formulate $\mathbf{C} \models x$ (with x a free parameter).

What we will don instead is to relativize a formula φ to a class $\mathbf{C}$, by replacing all $\forall v$ by $\forall v \in \mathbf{C}$ (and the same for $\exists$), resulting in a new formula called $\varphi^{\mathbf{C}}$. (In the case of the universal class V, $\varphi^{\mathbf{C}}$ is of course logically equivalent to φ.) So for a specific formula φ given in the meta language, we can formulate $\mathbf{C} \models \varphi$ by simply stating (in the object language) the new formula $\varphi^{\mathbf{C}}$. Note that we can do this for one (or finitely many) formulas onyl, we cannot quantify over φ in the object language. So in particular, it is completely trivial that for all (in the meta language!) $\varphi \in ZFC$ the following is provable in ZFC: φ^V (which is just φ, of course); but we can not formulate in ZFC "for all ZFC-axioms x, $V \models x$" (since there is no truth predicate).

Remark: We *can* of course formulate in ZF "there is a (set) M which is a ZF-model M", which is (by the completeness theorem) equivalent to Con(ZFC) and therefore (by the incompleteness theorem) not provable in ZF.

Definability of truth will be investigated in more detail in advanced set theory: It turns out that for a restricted (but infinite) class of formuals (e.g., Σ_n formulas) there is a truth predicate that works for this restricted class (and is itself a Σ_n formula).

Another important consequence of the fixed point lemma is Gödel's incompleteness theorem (Kunen 14.3): Let T be any recursive, consistent set of formulas containing ZF. Then in T we (can formulate but cannot prove) Con(T).