Definability Theory Course Notes Department of Logic, ELTE, Budapest, Spring semester 2014

Andréka, H. and Németi, I.

Aug.28, 2014

We treat definability theory as part of mathematical logic. We consider the subject of mathematical logic to be modeling (mathematically) our reasoning/thinking about the world. Definability theory is about structuring knowledge and about concept formation. Gradually, mathematical logic tries to model more and more aspects of reasoning.

Definitions are important in communication and in a precise, axiomatic thinking. What are definitions? Are they mere abbreviations, tools for ease of communication? Do they have a role in concept-formation, in the process of abstraction? We believe that they are essential in modern, axiomatic thinking.

Contents

1 Making definitions in First-order Logic FOL

What is a definition? When do we say that we indeed defined something? E.g., "give me the chalk that is on the table". We specified which chalk to give if there is exactly one chalk on the table. Otherwise you can reply: "I cannot give you the chalk because there is none on the table", or "There are more than one chalks on the table, please specify further which one you want to have", or "There are many chalks on the table, are any one of them suitable for you?". Here "$\mathsf{Onthetable}(x)$" is a definition (in this room) iff[1] $\mathsf{Thisroom} \models \exists! x \mathsf{Onthetable}(x)$.[2] In another room this may not be a good definition. Exactly the same thing happens when we prove from ZF set theory that there is a unique set which has no element, and then we say: "Let $\emptyset$ denote this set", and from here on we use $\emptyset$ as if it was a constant symbol in our language. We adjoined a new constant symbol $\emptyset$ to the language of set theory. This may be seen as a tool for an ease of communication, because, we can "eliminate" this new constant symbol: for any formula containing the symbol $\emptyset$ we can construct another formula of the old language, i.e., a formula which does not contain $\emptyset$ such that the two formulas mean the same

[1] iff means "if and only if"

[2] $\exists!$ means: exists one and only one: $\exists! x \varphi(x) \iff [\exists x \varphi(x) \wedge \forall x_1 x_2 (\varphi(x_1) \wedge \varphi(x_2) \to x_1 = x_2)]$.

thing in $\mathsf{ZF} \cup \{\forall x(x = \emptyset \leftrightarrow \neg\exists y\, y \in x)\}$. In set theory, we define the union of two sets, intersection of two sets, ordered pair of two sets the same style (i.e., $x \cup y, x \cap y, \langle x, y\rangle$). In no time, we use a rich language when talking about sets, in spite of the fact that the language of set theory contains only one binary relation symbol, $\in$.

The same style, we can define new relations. In the following, we concentrate on relations.[3] To define R, we say as much about a relation R that already makes this R unique. Note that we always need some background "knowledge" for this to happen, e.g., we rely on a theory[4] Th (later we will also talk about one given model in place of Th). Sometimes we will omit mentioning this background theory Th, but we always mean to have one in mind if we do not say otherwise. By a FOL language[5] we mean one with equality and perhaps with function and constant symbols. However, we will concentrate on relation symbols only.

Definition 1.1 *(definition of* R *in* Th*) Let* $\mathcal{L}$ *be a FOL language, let* Th *be a theory in* $\mathcal{L}$*, and let* Σ *be a theory in the language* $\mathcal{L}$ *extended with a new* n*-place relation symbol* R*. We call* Σ *a* description of R*. We say that* Σ defines R*, or* Σ *is a* definition *of* R *in* Th *iff for any model* $\mathfrak{M}$ *of* Th *there is exactly one* $R \subseteq M^n$ *such that* $\langle \mathfrak{M}, R\rangle \models \Sigma$*.* □

Note that when we define something, we give it a "name" R and a "meaning", or "specification" $\Sigma(\mathsf{R})$ to it.[6]

For example, this is how we define the unary function $\mathsf{factorial}$ in Peano Arithmetic PA.[7] The usual way of defining the factorial of n is: $\mathsf{factorial}(n) = 1 \cdot 2 \cdot \ldots \cdot n$ if $n \geq 1$ and 1 for $n = 0$. This is equivalent to the following.

$$\Sigma(\mathsf{factorial}) = \{\mathsf{factorial}(0) = 1, \quad \forall n \ \ \mathsf{factorial}(n+1) = \mathsf{factorial}(n) \cdot (n+1)\}$$

This Σ defines $\mathsf{factorial}$ in Peano Arithmetic PA, i.e., for every model $\mathfrak{M}$ of PA there is exactly one unary function $f : M \to M$ such that $\langle \mathfrak{M}, f\rangle \models \Sigma(f)$. For more examples see section 2.

[3] This is no serious restriction because we can treat an n-place function symbol to be a special $n+1$-place relation, see section 3.

[4] By a theory of $\mathcal{L}$ we simply mean a set of $\mathcal{L}$-formulas.

[5] FOL means First-order Logic.

[6] We write sometimes $\Sigma(\mathsf{R})$ in place of Σ, just to indicate that the symbol R can occur in Σ. Thus $\Sigma(\mathsf{R})$ and Σ denote the same thing. In such situations $\Sigma(\mathsf{R}')$ denotes the set of formulas we get from Σ by replacing R everywhere with R'.

[7] For the definition of PA see Def.2.2.

It is not always easy to see about a description $\Sigma(\mathsf{R})$ whether it is a definition or not, modulo a theory Th. There is a kind of description, though, when the form of the description ensures that it is a definition, this is called explicit definition.

Definition 1.2 *(explicit definition of* R*) Let* $\mathcal{L}$ *be a FOL language, and let* R *be an* n*-place relation symbol not present in* $\mathcal{L}$*. We say that* Σ *is an* explicit definition *of* R *iff* Σ *is of form* $\{\forall x_1, ..., x_n(\mathsf{R}(x_1, ..., x_n) \leftrightarrow \varphi)\}$ *for some* $\mathcal{L}$*-formula* φ *such that the free variables of* φ *are among* $x_1, ..., x_n$*. We also say that* Σ *is an explicit definition of* R *via* φ*.* □

To distinguish "ordinary" definitions from explicit ones, we will call the ordinary ones *implicit* definitions. Implicit definitions are sometimes more informative, more useful than explicit ones. In some sense, an implicit definition tells us "what properties make R what it is", while an explicit definition simply tells us "how we can construct R". Section 2 contains examples of this kind. Explicit definitions will be important for us when comparing two theories on different languages. The next theorem, called weak Beth definability theorem, states that the two notions of implicit and explicit definability coincide in FOL, modulo theories. (We will see in section 2 that these notions do not coincide modulo single structures.)

Theorem 1.1 *(weak Beth definability theorem) Let* $\mathcal{L}$ *be a FOL language, let* Th *be a theory in* $\mathcal{L}$*, and let* R *be a relation symbol not in* $\mathcal{L}$*. Each implicit definition of* R *is equivalent, modulo* Th*, to an explicit definition of* R*.*

Proof.[8] Assume that $\Sigma(\mathsf{R})$ defines R in Th, we will find an equivalent explicit definition for R. By our assumption that in each model of Th there is at most one relation satisfying Σ, we have[9]

$$\mathsf{Th} \cup \Sigma(\mathsf{R}) \cup \Sigma(\mathsf{R}') \models \forall \bar{x}(\mathsf{R}(\bar{x}) \leftrightarrow \mathsf{R}'(\bar{x})),$$

where R' is a new n-place relation symbol. Let $c_1, ..., c_n$ be new constant symbols, then we have

$$\mathsf{Th} \cup \Sigma(\mathsf{R}) \cup \Sigma(\mathsf{R}') \models \mathsf{R}(\bar{c}) \leftrightarrow \mathsf{R}'(\bar{c}).$$

[8] You can find basically this proof in [8, Thm.2.2.22].

[9] $\bar{x}$ denotes $x_1, ..., x_n$. We will use similar notation without warning.

By the compactness theorem, there are formulas $\theta \in \mathcal{L}$ and $\sigma(\mathsf{R})$ such that $\mathsf{Th} \models \theta$ and $\Sigma(\mathsf{R}) \models \sigma(\mathsf{R})$ and $\theta \wedge \sigma(\mathsf{R}) \wedge \sigma(\mathsf{R}') \models \mathsf{R}(\bar{c}) \leftrightarrow \mathsf{R}'(\bar{c})$. Moving R on one side of $\models$, and R' on the other, we get

$$\theta \wedge \sigma(\mathsf{R}) \wedge \mathsf{R}(\bar{c}) \models \sigma(\mathsf{R}') \rightarrow \mathsf{R}'(\bar{c}). \tag{1}$$

Now we will use Craig's Interpolation Theorem, later we will prove it (see Thm.1.3). From Craig's Interpolation Theorem we get an interpoland $\varphi(\bar{c})$ in the language $\mathcal{L}$ expanded with the constants such that

$$\theta \wedge \sigma(\mathsf{R}) \wedge \mathsf{R}(\bar{c}) \models \varphi(\bar{c}), \quad \varphi(\bar{c}) \models \sigma(\mathsf{R}') \rightarrow \mathsf{R}'(\bar{c}). \tag{2}$$

Since R' does not occur in φ, the latter part of equation (2) is equivalent to

$$\varphi(\bar{c}) \models \sigma(\mathsf{R}) \rightarrow \mathsf{R}(\bar{c}). \tag{3}$$

From equations (2), (3) we get that

$$\theta \wedge \sigma(\mathsf{R}) \models \mathsf{R}(\bar{c}) \leftrightarrow \varphi(\bar{c}), \quad \text{so} \tag{4}$$

$$\theta \wedge \sigma(\mathsf{R}) \models \forall \bar{x}(\mathsf{R}(\bar{x}) \leftrightarrow \varphi(\bar{x})). \tag{5}$$

Thus, by the properties of θ, σ we have[10]

$$\mathsf{Th} \cup \Sigma(R) \models \mathsf{R} \leftrightarrow \varphi. \tag{6}$$

Now, using that in each model of Th there exists a relation satisfying Σ, equation (6) implies $\mathsf{Th} \models \Sigma(\mathsf{R}/\varphi)$, and so

$$\mathsf{Th} \cup \{\mathsf{R} \leftrightarrow \varphi\} \models \Sigma(\mathsf{R}). \tag{7}$$

Finally, (6),(7) state that $\Sigma(\mathsf{R})$ and $\mathsf{R} \leftrightarrow \varphi$ are equivalent modulo Th; and $\mathsf{R} \leftrightarrow \varphi$ is an explicit definition. □

Why is the previous theorem called "weak Beth definability theorem"? The reason is that there is a "stronger" version, and that stronger version was first proved by Everett Beth in 1953. We now state the original Beth theorem, too, its proof is analogous to the above one.

[10]We will write $\mathsf{R} \leftrightarrow \varphi$ for $\forall \bar{x}(\mathsf{R}(\bar{x}) \leftrightarrow \varphi)$, we may use similar abbreviations without mentioning.

Definition 1.3 *(weak definitions) We say that* $\Sigma(\mathsf{R})$ weakly implicitly *defines* R *in* Th *if in each model of* Th *there is* at most *one relation* R *that satisfies* $\Sigma(\mathsf{R})$. *We say that* $\Sigma(\mathsf{R})$ weakly explicitly *defines* R *in* Th *if there is a* φ *in the language of* Th *such that* $\mathsf{Th} \cup \Sigma(\mathsf{R}) \models \mathsf{R} \leftrightarrow \varphi$.

Theorem 1.2 *((strong) Beth definability theorem)* $\Sigma(\mathsf{R})$ *is a weak implicit definition iff it is a weak explicit one (modulo any theory* Th*).*

The proof of Theorem 1.2 is completely analogous to the proof of Theorem 1.1, we leave this as an exercise.

Remark 1.1 A weak implicit definition does not require the existence of the defined relation in each model while an ordinary implicit definition does (compare Def.1.1). This is the reason for the adjective "weak" in weak definition. There are more weak definitions than strong ones. And this is the reason for the adjective "strong" in strong Beth definability theorem: it states existence of an equivalent explicit definition for more implicit definitions (than the weak Beth theorem).

Chang-Keisler [8] states only the strong Beth theorem and, because of this, implicit and explicit definitions are defined in [8], and in many other books, as our weak corresponding notions. The (strong) Beth definability theorem has more consequences, but the weak Beth theorem is more intuitive. The weak Beth definability property is widely used, e.g., in abstract model theory [3]. □

We note that Craig's interpolation theorem, Theorem 1.3 below, belongs to definability theory not only because it is used in the proof of Beth definability theorem. It is a typical definability theorem in that it concerns the relationship between languages of different vocabularies.

Theorem 1.3 *(Craig's Interpolation Theorem) Let* φ, ψ *be FOL formulas and assume* $\varphi \models \psi$. *Then there is a formula* θ *in the common language of* φ, ψ *such that* $\varphi \models \theta$ *and* $\theta \models \psi$.

Example 1.1 *We can check that* $\exists x \mathsf{R}(x) \wedge \exists x \neg \mathsf{R}(x) \models [(\exists x \mathsf{B}(x) \wedge \exists x \neg \mathsf{B}(x)) \vee \exists xy(\mathsf{B}(x) \wedge \mathsf{B}(y) \wedge x \neq y) \vee \exists xy(\neg \mathsf{B}(x) \wedge \neg \mathsf{B}(y) \wedge x \neq y)]$. *A good interpoland here is* $\exists xy\, x \neq y$.

Proof. Let the languages of φ and ψ be $\mathcal{L}_1$ and $\mathcal{L}_2$, respectively, and let $\mathcal{L}$ denote the intersection of these two languages. Define $T_0 = \{\theta \in \mathcal{L} : \varphi \models \theta\}$. We have to prove that $T_0 \models \psi$. (This is enough to prove because by the compactness theorem $T_0 \models \psi$ implies $\Gamma \models \psi$ for some finite $\Gamma \subseteq T_0$. We can choose Γ to be nonzero because we have equality in the language and then we can take the interpoland to be $\bigwedge \Gamma$.)

Assuming that $T_0 \not\models \psi$, we are going to construct a model for $\varphi \wedge \neg\psi$, which contradicts our original assumption $\varphi \models \psi$. Let $\mathfrak{M}$ be a model of $T_0 \cup \{\neg\psi\}$. There is such a model by our assumption $T_0 \not\models \psi$. Let Th_0 be the set of all $\mathcal{L}$-formulas valid in this model (in other words, Th_0 is the theory of the $\mathcal{L}$-reduct of $\mathfrak{M}$),[11] i.e.,

$$\mathsf{Th}_0 = \{\theta \in \mathcal{L} : \mathfrak{M} \models \theta\}.$$

We want to show that $\mathsf{Th}_0 \cup \{\varphi\}$ is consistent. Indeed, if it was not, θ, φ would be inconsistent for some $\theta \in \mathsf{Th}_0$ by the compactness theorem, which means that $\varphi \models \neg\theta$. But then $\neg\theta \in T_0 \subseteq \mathsf{Th}_0$, contradicting $\theta \in \mathsf{Th}_0$, $\mathfrak{M} \models \mathsf{Th}_0$. Thus we have

$$\mathsf{Th}_0 \cup \{\varphi\} \quad \text{is consistent.}$$

Let $\mathfrak{N}$ be a model of $\mathsf{Th}_0 \cup \{\varphi\}$. Let $\mathfrak{M}', \mathfrak{N}'$ be the reducts of these two models to the common language $\mathcal{L}$. If $\mathfrak{M}'$ and $\mathfrak{N}'$ happened to be the same, then "the union of $\mathfrak{M}$ and $\mathfrak{N}$" would be a model, and in this model φ and $\neg\psi$ would be true, finishing the proof.

We may not assume that $\mathfrak{M}', \mathfrak{N}'$ are equal, but we know that the same formulas are true in them, by $\mathfrak{N} \models \mathsf{Th}_0$. Now we will use "heavy machinery" to finish the proof, but after that we will give also an elementary proof for this last step. By the Keisler-Shelah ultraproduct theorem[12], $\mathfrak{M}'$ and $\mathfrak{N}'$ have isomorphic ultrapowers. Let $\mathfrak{M}$" and $\mathfrak{N}$" be the corresponding ultrapowers of the "richer" models $\mathfrak{M}$ and $\mathfrak{N}$. Now, $\mathfrak{M}$" $\models \neg\psi$, $\mathfrak{N}$" $\models \varphi$ (by $\mathfrak{M} \models \neg\psi$, $\mathfrak{N} \models \varphi$) and the $\mathcal{L}$-reducts of $\mathfrak{M}$" and $\mathfrak{N}$" are isomorphic. Then we may assume that the $\mathcal{L}$-reducts of $\mathfrak{M}$" and $\mathfrak{N}$" are the same and we can unite $\mathfrak{M}$" and $\mathfrak{N}$" into one model of the language $\mathcal{L}_1 \cup \mathcal{L}_2$ in which $\varphi \wedge \neg\psi$ is true, and we are done. We return to a more elementary proof below. □

[11] A reduct of a model is the same model without some of its functions and/or relations.

[12] See [8, Thm.6.1.15, Isomorphism Theorem].

More elementary proof for the last step. We want to modify the above idea so that in place of the Keisler-Shelah ultraproduct theorem we use more elementary methods. This can be done as follows.

We want to construct two models, $\mathfrak{M}$ and $\mathfrak{N}$ simultaneously, step-by-step, such that $\mathfrak{M} \models \neg\psi$, $\mathfrak{N} \models \varphi$, and their $\mathcal{L}$-reducts are the same. To achieve this goal, we add countably many new constant symbols $c_0, c_1, ...$ to the language, we want to construct our models $\mathfrak{M}$ and $\mathfrak{N}$ so that their universes be the set C of these constants (maybe factorized by an equivalence relation). Let $\mathcal{L}_1^+$ denote the language $\mathcal{L}_1$ together with these constants, and similarly let $\mathcal{L}_2^+$, $\mathcal{L}^+$ denote the languages $\mathcal{L}_2$, $\mathcal{L}$ together with these constants. To define the model $\mathfrak{M}$ amounts deciding whether $\mathsf{B}(c_i, c_j)$ holds for each binary relation symbol B and constants c_i, c_j in $\mathfrak{M}$, and the same for other basic relation symbols of $\mathcal{L}_2$. We also want $\neg\psi$ be true in $\mathfrak{M}$. Since $\neg\psi$ may imply that its models be finite, say have exactly 5 elements, we may need to construct a finite model $\mathfrak{M}$, for this reason we also will decide the formulas of form $c_i = c_j$ (for $i, j < \omega$).[13] Our plan for constructing $\mathfrak{M}$ is to list all formulas of $\mathcal{L}_2^+$, and we will "decide" them one after the other so that we always keep in mind our previous decisions (i.e., the n-th decision should be consistent with all the previous decisions).

In order that all the formulas we "decide" positively should indeed hold in the model $\mathfrak{M}$ we construct, we make sure that whenever we decide for a formula of form $\exists x \chi(x)$ to hold, we also decide which x should satisfy this formula, i.e., we add to our decision $\chi(c_k)$ for some k. We call this step "*constant-filling step*". This latter idea and the whole method of constructing a model this way originates with Leon Henkin, sometimes this is called "the Henkin-method for constructing a model". This method is described in detail in [8, beginning of section 2.1].

Keeping the above in mind, here is the strategy for constructing models $\mathfrak{M}$ and $\mathfrak{N}$ of languages $\mathcal{L}_2$ and $\mathcal{L}_1$ respectively. Our "wishes" are: $\mathfrak{M} \models \neg\psi$, $\mathfrak{N} \models \varphi$, and their $\mathcal{L}$-reducts be the same. The latter in the present case can be achieved by requiring that they satisfy the same $\mathcal{L}^+$-formulas, because their universes will be built out of the constants, therefore their whole structures will be coded in their $\mathcal{L}^+$-theories. Our "tool" is: we know that there is no interpoland between φ and ψ.

List all the formulas in $\mathcal{L}_2^+$ such that $\neg\psi$ is the first formula, let $\psi_0, \psi_1, ...$ be this list. Do the same for the $\mathcal{L}_1^+$ formulas such that φ is the first one

[13] ω denotes the set of natural numbers $\{0, 1, ...\}$.

in the list, let $\varphi_0, \varphi_1, ...$ be this list. We will decide alternately from the two lists, such that our first decision in the ψ_i list will be "hold" for $\neg\psi$, and the first decision in the φ_i list will be "hold" for φ. Then we go on, step by step deciding each formula on the two lists.

In the n-th step let $\psi'_0, \psi'_1, ..., \psi'_{n-1}$ represent our decisions so far (i.e., ψ'_j is ψ_j if we decided ψ_j to hold, and let ψ'_j be $\neg\psi_j$ if we decided ψ_j not to hold). Let Ψ_n be the conjunction of all these, and let $\varphi'_0, \varphi'_1, ..., \Phi_n$ be the analogous things for the φ_i list. Then

$$\Psi_n \text{ and } \Phi_n \text{ represent our previous choices, and} \quad \Psi_0 \text{ is } \neg\psi, \quad \Phi_0 \text{ is } \varphi.$$

We want to maintain that Ψ_n and Φ_n do not contradict on the common language $\mathcal{L}^+$, i.e.,

$$\Phi_n \models \theta \text{ together with } \Psi_n \models \neg\theta \text{ holds for no } \theta \in \mathcal{L}^+.$$

In the first two steps of this construction, the above inductive hypothesis will hold because of our "tool" we have: if $\varphi \models \theta$ and $\neg\psi \models \neg\theta$ for some $\theta \in \mathcal{L}^+$, then $\varphi \models \theta \models \psi$, and then the formula $\forall\bar{x}\theta(\bar{x}/\bar{c})$ in which we replace the constants with new variables will be an interpoland, which we know we do not have. We also have to show that in the n-th step either ψ_n or $\neg\psi_n$ will be a good choice (and the same for the other list). Here we can use the inductive hypothesis that Φ_{n-1} and Ψ_{n-1} do not contradict on the common language. In each case when our decision is of form $\exists x\chi(x)$ we immediately choose a constant c not occurring in our so-far-made decisions and add the decision $\chi(c)$ to our decisions made so far. For the details see [8, proof of Thm.2.2.20]. □

Craig's interpolation theorem is true for FOL without equality, but in this case we have to add a formula $\bigwedge\emptyset$ to the language. This last formula usually is denoted by $\bot$ or by FALSE, and then usually the dual formula $\top$ (or TRUE) is added to the language, too. We will talk about the equality-free languages in section 3. Until then: the importance of the equality-free FOL is that it *explains* equality.

2 Examples

2.1 The role of infinite models

Example 2.1 *Let* $\mathfrak{N}(100) = \langle \{n \in \omega : n \leq 100\}, suc \rangle$ *where* $suc(n) = n + 1$ *if* $n < 100$ *and* $suc(100) = 100$. *Define the set* E *of even numbers in it! Explicitly. Give a shorter implicit definition.*

Here is an explicit definition for E which works in $\mathfrak{N}(100)$: $x = 0 \vee x = 2 \vee ... \vee x = 100$. Well, $0, 2, ..., 100$ are not in the language of $\mathfrak{N}(100)$, hence we replace them with some of their explicit definitions. E.g., 0 is the unique element which is not a successor of any other element, then 2 is $\mathsf{suc}(\mathsf{suc}(0))$, etc. Here is an explicit definition for E with these written in:

$$\exists z(\neg\exists y(z = \mathsf{suc}(y)) \wedge (x = z \vee x = \mathsf{suc}(\mathsf{suc}(z)) \vee ... \vee x = \mathsf{suc}^{100}(z)))$$

The above is a formula $\varphi(x)$ on the language of $\mathfrak{N}(100)$ with x as the only free variable, and

$$\mathfrak{N}(100) \models \varphi(n) \quad \text{iff} \quad n \in E.$$

This explicit definition is quite a long formula. Here is an implicit definition $\Delta(\mathsf{E})$ which works for E in $\mathfrak{N}(100)$:

$$\mathsf{E}(0) \wedge \forall z\, z \neq \mathsf{suc}(z) \rightarrow [(\mathsf{E}(z) \rightarrow \neg\mathsf{E}(\mathsf{suc}(z))) \wedge (\neg\mathsf{E}(z) \rightarrow \mathsf{E}(\mathsf{suc}(z)))].$$

Of course, 0 has to be replaced with its definition as above. We can see that this implicit definition of E is shorter, and it is not a mere enumeration of the elements of E but it really describes it in a way by saying that E is defined by selecting every second element. Let k be any number and define $\mathfrak{N}(k)$ by replacing 100 in the definition of $\mathfrak{N}(100)$ with k (i.e., $\mathfrak{N}(k) = \langle \{n \in \omega : n \leq k\}, \mathrm{suc} \rangle$ where $\mathrm{suc}(n) = n + 1$ if $n < k$ and $\mathrm{suc}(k) = k$). Now, the set of even numbers in $\mathfrak{N}(k)$ can be defined with the same implicit definition as in $\mathfrak{N}(100)$, but its explicit definition will be a bit longer formula since we have to list a different number of elements.

Let us try to do the same in $\mathfrak{N}(\omega)$ (defined analogously)! The implicit definition $\Delta(\mathsf{E})$ of the even numbers works even here, i.e., the set E of even numbers is the unique (unary) relation on ω that satisfies $\Delta(\mathsf{E})$ in $\mathfrak{N}(\omega)$. Is there an explicit definition for E in $\mathfrak{N}(\omega)$? The one analogous to the explicit definitions we used above does not work, because we have to list infinitely many elements, but a formula can list only finitely many elements. So, is there an explicit definition of the even numbers in $\mathfrak{N}(\omega)$ at all?

Theorem 2.1 *The set E of even numbers cannot be defined explicitly in $\mathfrak{N}(\omega)$. I.e., for each formula $\varphi(x)$ in the language of $\mathfrak{N}(\omega)$ with one free variable x we have that $E \neq \{n \in \omega : \mathfrak{N}(\omega) \models \varphi(n)\}$.*

Proof. Add infinitely many constants c_k to the language of $\mathfrak{N}(\omega)$ and let $\mathfrak{N}(\omega)^+$ denote the expanded structure in which c_k denotes k, for every $k \in \omega$. Now, add one more new constant symbol c to this language and write up the following theory Th:

$$\mathsf{Th}(\mathfrak{N}(\omega)^+) \cup \{c \neq c_k : k \in \omega\}$$

It is easy to see that every finite subset of Th is consistent: let T_0 be any finite subset of Th, then finitely many constants occur in it. Let $k \in \omega$ be such that no constant c_i with $i \geq k$ occurs in T_0. Let $\mathfrak{N}'$ be the model we obtain from $\mathfrak{N}(\omega)^+$ such that we expand it with c denoting k. Then $\mathfrak{N}' \models T_0$. By the compactness theorem then Th has a model $\mathfrak{M}$. Since the theory of successor is contained in Th, this $\mathfrak{M}$ consists of one island like ω and some islands like the whole numbers with successor. More precisely, the unary function suc on the constants c_k in $\mathfrak{M}$ behaves exactly as in $\mathfrak{N}(\omega)$. The interpretation of c in $\mathfrak{M}$, is not in "this island", and suc on this element behaves as in $\mathbb{Z}$, where $\mathbb{Z}$ denotes the set of whole numbers, i.e., the elements $\mathsf{suc}^z(c)$ are all distinct for $z \in \mathbb{Z}$.

Let us consider the function $f : M \longrightarrow M$ which is identity everywhere, except on the above-mentioned island $\mathbb{Z}$ which it shifts upward by one, i.e., let us define $f(\mathsf{suc}^z(c)) = \mathsf{suc}^{z+1}(c)$ for elements of this form, and let us define $f(a) = a$ for all other elements $a \in M$. Let $\mathfrak{M}'$ be the reduct of $\mathfrak{M}$ where we "forget" the constants[14]. Then f is an automorphism of $\mathfrak{M}'$ (i.e., it is a permutation of M which respects suc). This shows that all the elements in the "island of c" are completely alike, hence they satisfy the same formulas. I.e., it can be shown (e.g., by induction) that

$$\mathfrak{M}' \models \varphi(n) \quad \text{iff} \quad \mathfrak{M}' \models \varphi(f(n))$$

for all $n \in M$ and formula $\varphi(x)$.

Assume now that $\varphi(x)$ would define E in $\mathfrak{N}(\omega)$. Then

$$\mathfrak{N}(\omega) \models \forall x(\varphi(x) \leftrightarrow \neg\varphi(\mathsf{suc}(x))).$$

[14]We used the constants only for ensuring that $\mathfrak{M}'$ is a real extension of our $\mathfrak{N}(\omega)$

So, the latter formula is in $\mathsf{Th}(\mathfrak{N}(\omega))$ and hence it is also true in $\mathfrak{M}'$. But we have just seen that no such thing can be true in the island of c (because of the automorphism f). $\square$

We have seen that the set of even numbers can be implicitly defined in $\mathfrak{N}(\omega)$ while it cannot be explicitly defined in it. This shows that implicit and explicit definitions have the same power only modulo a theory, and not modulo single structures.

The present example also shows that Beth theorem does not hold for finite model theory. Let $\mathrm{FOL}^{<\omega}$ denote first-order logic with only finite models.

Theorem 2.2 *Beth definability theorem does not hold for $FOL^{<\omega}$: there is a theory Th and a description $\Delta(\mathsf{R})$ that has a unique solution in each finite model of Th, yet there is no explicit definition of R that works in each finite model of Th.*

Proof. Let $\mathsf{K} = \{\mathfrak{N}(k) : k \in \omega\}$. Then it is easy to show that any finite model of $\mathsf{Th}(\mathsf{K})$ is[15] isomorphic to a member of K. Thus $\Delta(\mathsf{R})$ is an implicit definition (of the even numbers) in $\mathsf{Th}(\mathsf{K})$. We are going to show that $\Delta(\mathsf{E})$ is not equivalent to any explicit definition in $\mathsf{Th}(\mathsf{K})$.

Let $\mathsf{Th} = \mathsf{Th}(\mathsf{K}) \cup \{c \neq \mathsf{suc}^n(0), \mathsf{suc}^n(c) \neq \mathsf{suc}^{n+1}(c) : n \in \omega\}$ where c is a new constant. Then this Th is consistent because each of its finite subsets is consistent. (In more detail: let T_0 be a finite subset of Th. Then $T_0 \subseteq \mathsf{Th}(\mathsf{K}) \cup \{c \neq \mathsf{suc}^n(0), \mathsf{suc}^n(c) \neq \mathsf{suc}^{n+1}(c) : n \leq k\}$ for some $k \in \omega$. Then T_0 is true in $\mathfrak{N}(2k+2)$ expanded with c to denote $k+1$.) Thus Th has a model, let $\mathfrak{N}$ be an arbitrary model of Th. This $\mathfrak{N}$ is infinite, and the "island of c" is like $\mathbb{Z}$. From here on we can use the proof of Theorem 2.1 to show that the set of even numbers in models of $\mathsf{Th}(\mathsf{K})$ cannot be defined explicitly. $\square$

The above theorem shows that infinite models are needed for testing existence of explicit definitions. Even when we are interested in one specific model, or we are interested in finite models only, infinite models are useful for showing whether a concrete implicit definition can or cannot be written into an explicit one. They serve as "indicators" for non-existence of explicit definitions. In this respect, they can be taken as "nonstandard models" for finite model theory.

[15] $\mathsf{Th}(\mathsf{K})$ denotes the set of all formulas valid in K.

2.2 Why Peano Arithmetic?

We turn to inspecting recursive definitions and why Peano's axioms are the way they are. In the implicit definitions below we do not write out the universal quantifiers in order to make the formulas more readable.[16]

Example 2.2 *Let* $\Delta(+) = \{0 + x = x, \quad \mathsf{suc}(y) + x = \mathsf{suc}(y + x)\}$.

The above $\Delta(+)$ is a so-called recursive definition of $+$ from suc. Recall that 0 is definable from suc as the unique number which is not a successor of any number. Thus $\Delta(+)$ written out fully is

$$\{\forall xz(\neg\exists y\, z = \mathsf{suc}(y) \rightarrow z + x = x), \forall yx\, \mathsf{suc}(y) + x = \mathsf{suc}(y + x)\}.$$

This $\Delta(+)$ defines $+$ as a kind of iteration of suc:

$$y + x = \underbrace{\mathsf{suc}...\mathsf{suc}}_{y\ times}(x) = \mathsf{suc}^y(x)\ .$$

This is an implicit definition of $+$ that works in $\langle\omega, \mathsf{suc}\rangle$, i.e., there is exactly one binary function, namely addition, that "satisfies" $\Delta(+)$ in $\langle\omega, \mathsf{suc}\rangle$. Theorem 2.1 implies that there is no explicit definition that would be equivalent to $\Delta(+)$ (because then the set of even numbers could be explicitly defined by the formula $\exists y\, x = y + y$). Beth definability theorem (Theorem 1.2) then implies that $\Delta(+)$ is not even a weak implicit definition in $\mathsf{Th}(\langle\omega, \mathsf{suc}\rangle)$, i.e., there is a model which is elementarily equivalent[17] to $\langle\omega, \mathsf{suc}\rangle$ and in which there are at least two solutions for $\Delta(+)$. We note that suc can be defined from $+$ explicitly.

Example 2.3 *Let* $\Delta(\star) = \{0 \star x = 0, \quad \mathsf{suc}(y) \star x = y + (y \star x)\}$.

The above $\Delta(\star)$ is a recursive definition of $\star$ from $\mathsf{suc}, +$. It defines $\star$ as an iteration of $+$:

$$y \star x = \underbrace{x + ... + x}_{y\ times}\ .$$

[16] In this we keep to the convention that validity of an open formula in a model is defined as validity of the universal closure of the formula in question.

[17] Two models are said to be elementarily equivalent if they are not distinguishable by a formula, i.e., if their theories are the same.

This $\Delta(\star)$ is an implicit definition of $\star$ in $\langle \omega, + \rangle$, i.e., there is exactly one binary function, namely multiplication, that "satisfies" $\Delta(\star)$ in it. Can this $\star$ be defined from $+$ explicitly?

The answer is the same as in the first example: no, it cannot be defined explicitly. However, there is a bigger "jump" in the expressive power when we "add" multiplication to our language than when we add addition to successor, because of the following. Both the theory of successor and the theory of addition (called Presburger Arithmetic) are decidable[18]. These facts can be proved by the so-called elimination of quantifiers method. However, the theory of addition and multiplication (called Arithmetic) is not only not decidable, but it is not even recursively enumerable[19]. Above, by "theory of successor" we mean $\mathsf{Th}(\langle \omega, \mathsf{suc} \rangle)$, and similarly for the other theories mentioned.

Why is this big jump here in complexity when we add multiplication? What bigger jump in expressive power can we expect when we add the iteration of multiplication, i.e., exponentiation, to the language?

Example 2.4 *Let* $\Delta(\mathsf{exp}) = \{x^0 = 1, \quad x^{\mathsf{suc}(y)} = x \star x^y\}$*, where* x^y *denotes* $\mathsf{exp}(x, y)$ *and* 1 *is the unique number for which* $1 \star x = x$ *for all* x*.*

As before,

$$x^y = \underbrace{x \star ... \star x}_{y \text{ times}} .$$

This $\Delta(\mathsf{exp})$ is an implicit definition in $\langle \omega, +, \star \rangle$. Can exponentiation defined explicitly in this structure? The answer this time is affirmative: yes, exponentiation can be explicitly defined from addition and multiplication (we are going to show an explicit definition). So, instead of having here an even bigger jump, we have no jump at all.

The reason both for the "big jump" between addition and addition+multiplication, and for the "no jump" between addition+multiplication and addition+multiplication+exponentiation is that we can express finite sequences once we have addition and multiplication. We note that multiplication in itself is not strong, the theory of multiplication is decidable (theorem of

[18]This means that one can write a computer program which, taken a formula as input, halts in finite steps and gives a "yes" answer iff the input is in the theory.

[19]There is no computer program that would start printing formulas one after the other (one formula in each step) so that it prints only formulas in the theory, and each formula in the theory gets printed out in one of the steps.

Mostowski, see the Feferman-Vaught paper [11]). Thus the strength comes from the interaction between addition and multiplication.

We now turn to showing how expressibility of finite sequences ensures expressibility of recursive definitions such as the definition of exponentiation. Assuming that we can express somehow sequences, first we show how these boost the power of explicit definitions, and then we turn to expressing (defining explicitly, sometimes called "coding") sequences. Using notions connected with sequences, here is an explicit definition of exponentiation:[20]

$z = x^y \leftrightarrow$

$\exists s\,[\mathsf{fin\text{-}seq}(s) \ \wedge \ y \leq \mathsf{length}(s) \ \wedge \ s_0 = 1 \ \wedge \ \forall i < y\, s_{i+1} = x \star s_i \ \wedge \ s_y = z].$

In order to convert the above into an explicit definition, we have to replace $\mathsf{fin\text{-}seq}$(s), length(s), and s_i with concrete formulas that express what we have in mind.

First we express pairs. By using pairs, we can code n-tuples for any fixed n the usual way[21], but we do not have a uniform finite way (formula) for reaching the i-th member of an n-tuple defined this way. We need this uniform formula, because we need to talk about the i-th member of a sequence where i is a variable. Therefore, by the help of pairs, we define sequences proper not as n-tuples but by using a different idea. Below, $\mathsf{mem}(s, i, a)$ stands for "the i-th member of sequence s is a", and $\mathsf{rem}(x, y, z)$ stands for "the remainder when dividing x with y is z".

Definition 2.1 *(pair, sequences)*

$\mathsf{pair}(x, y) = (x + y) \star (x + y) \ + \ y.$

$\mathsf{rem}(x, y, a) \leftrightarrow a < y \wedge \exists w\, x = w \star y + a.$

$\mathsf{mem}(s, i, a) \leftrightarrow \exists xy[s = \mathsf{pair}(x, y) \wedge \mathsf{rem}(x, 1 + (1 + i) \star y, a)].$ □

We turn to showing that the above definitions express what their names suggest.

[20] Ordering usually is defined from addition as $x \leq y \leftrightarrow \exists z(x + z = y)$, and $x < y \leftrightarrow x \leq y \wedge x \neq y$).

[21] 3-tuples $\langle a, b, c\rangle$ are defined as $\langle a, \langle b, c\rangle\rangle$, etc

Theorem 2.3 *The following formulas are true in* $\mathbb{N} := \langle \omega, +, \star \rangle$.

$\mathsf{pair}(x, y) = \mathsf{pair}(x', y') \to (x = x' \wedge y = y')$.

$x = \mathsf{pair}(y, z) \to \forall i \exists! a \, \mathsf{mem}(x, i, a)$.

$\forall a_0, ..., a_n \exists s \, (\mathsf{mem}(s, 0, a_0) \wedge \cdots \wedge \mathsf{mem}(s, n, a_n))$.

We do not prove the above theorem, for a proof we refer to [10, Lemma 37A, pp.246-249]. □

Lets say that a number x is a pair if $x = \mathsf{pair}(y, z)$ holds (in $\mathbb{N}$) for some y, z. The first formula in Theorem 2.3 expresses that taking out the first component of a pair is a function, and similarly for taking out the second component is a function. Thus, if a number is a pair of two numbers, these two numbers can be recovered from it. This justifies saying that x *codes* the pair $\langle y, z \rangle$ if $x = \mathsf{pair}(y, z)$. The second formula in Theorem 2.3 states that each pair "codes" an infinite sequence by the convention embodied in the formula mem.[22] Namely, if $x = \mathsf{pair}(y, z)$ then x codes the sequence $s = \langle s_0, s_1, \ldots, s_i, \ldots \rangle$ where s_i is the remainder when dividing y with $1+(1+i)\star z$. We will often use in the sequel that each pair codes an infinite sequence. Not all infinite sequences are coded by numbers this way, since there are more infinite sequences than numbers.[23] However, the third statement of Theorem 2.3 says that each finite sequence has an extension which is coded by a number via the convention mem.

We now can write up an explicit definition of exponentiation by substituting these notions in our earlier formula.

Theorem 2.4 *The formula below is an explicit definition of exponentiation in* $\mathbb{N}$*, i.e., the following is true in* $\mathbb{N}$*:*

$z = x^y \leftrightarrow$

$\exists s \, [\mathsf{mem}(s, 0, 1) \wedge \forall i < y \, \forall v (\mathsf{mem}(s, i, v) \to \mathsf{mem}(s, i + 1, x \star v)) \wedge \mathsf{mem}(s, y, z)]$.

Proof. Let x, y, z be natural numbers and let $\varphi(x, y, z)$ denote the formula of which we claim that it is an explicit definition of exponentiation in $\mathbb{N}$,

[22] Note that coding always implies the existence of a "key" by the use of which we can recover the coded thing from the code.

[23] They have different cardinalities in the set theoretical sense.

and assume that $z = x^y$. We have that $z = x^y$ iff the last member of the y-long sequence $\langle 1, x, x \star x, x \star x \star x, ..., x \star ... \star x \rangle$ is z, since exponentiation is defined as the iteration of $\star$. Let s be any number such that $\mathsf{mem}(s, 0, 1)$, $\mathsf{mem}(s, 1, x)$, $\mathsf{mem}(s, 2, x \star x)$, $\mathsf{mem}(s, 3, x \star x \star x)$, ... $\mathsf{mem}(s, y, z)$. There is such a sequence by the second formula in Theorem 2.3. Then this s satisfies $\mathsf{mem}(s, 0, 1) \ \wedge \ \forall i < y \forall v(\mathsf{mem}(s, i, v) \rightarrow \mathsf{mem}(s, i + 1, x \star v)) \ \wedge \ \mathsf{mem}(s, y, z)$, hence φ holds for our x, y, z. Conversely, assume that we have a sequence s that satisfies the $\exists s$-free part of φ. Let a_i denote the unique number for which $\mathsf{mem}(s, i, a_i)$ holds, for each $i < \omega$. By our statement then the sequence $\langle a_0, ..., a_y \rangle$ represents a good "computation" of x^y with the "output" z. Thus we indeed have $x^y = z$. □

The above explicit definition of exponentiation works not only in $\mathbb{N}$, but also in each model of Peano Arithmetic PA. Let us include the definition of PA here. The intended (or standard) model of PA is $\mathbb{N}$. Hence the language of PA contains two binary function symbols $+, \star$. Below we also use suc, since it can be defined from $+$.

Definition 2.2 *(Peano Arithmetic* PA*) Peano Arithmetic* PA *is defined as the set of all formulas listed below.*

$\exists! z \neg \exists y \, z = \mathsf{suc}(y)$. *Let* 0 *denote this unique element.*

$\forall xy(\mathsf{suc}(x) = \mathsf{suc}(y) \rightarrow x = y)$,

$0 + x = x, \quad \mathsf{suc}(y) + x = \mathsf{suc}(x + y)$,

$0 \star x = 0, \quad \mathsf{suc}(y) \star x = x + (y \star x)$,

$[\varphi(0) \wedge \forall x(\varphi(x) \rightarrow \varphi(\mathsf{suc}(x)))] \rightarrow \forall x \varphi(x)$,
for each formula φ *in the language of* $\mathbb{N}$ *where* φ *may contain any number of free variables.* □

All the so-called recursive definitions[24] can be turned into explicit definitions that work in PA. The idea is that the recursive definition delineates an algorithm for computing the result of the function, and this computation can be coded by a finite sequence. We note that PA is a weaker theory than $\mathsf{Th}(\mathbb{N})$ because the former is recursively enumerable (since PA is such) while

[24]For what recursive definitions are see, e.g., [9].

the latter is not recursively enumerable. Expressibility of all recursive functions is the reason why PA is such an often-used theory in spite that it is weaker than Arithmetic. Also, we now know why the definitions of addition and multiplication are included into PA and no further definitions such as the definition of exponentiation are included into PA.

2.3 Undefinability of truth

Are all implicit definitions in $\mathbb{N}$ equivalent to explicit ones in $\mathbb{N}$? The answer is in the negative, because of the following. Each element of $\mathbb{N}$ can be explicitly defined (as a constant) and so each subset of $\mathbb{N}$ can be implicitly defined (as a unary relation).[25] There are uncountably many subsets of $\mathbb{N}$ but there are only countably many formulas–hence explicit definitions–, so there are implicit definitions that work in $\mathbb{N}$ but have no equivalent explicit definitions.

However, in FOL each implicit definition is equivalent to a *finite* one, by the Beth Definability Theorem Thm.1.2 and the compactness theorem for FOL. Thus in FOL we can require implicit definitions to be finite, and nothing important would change. There are only countably many finite implicit definitions in a language with finitely many basic symbols. In view of this, the real question for $\mathbb{N}$ is whether all finite implicit definitions in it are equivalent to explicit ones. The answer to this question is also in the negative.

Next we present a finite implicit definition which works in $\mathbb{N}$ yet is equivalent to no explicit definition in $\mathbb{N}$. This example will exploit Tarski's theorem on undefinability of truth, leading up to showing that weak second-order logic (wSOL) does not have the Beth Definability Property (BDP).[26] With this example we also want to illustrate how using expressibility of sequences can be used almost like programming any idea that we have clearly enough in our minds.

The finite implicit definition in $\mathbb{N}$ that is not equivalent to any explicit one in $\mathbb{N}$ will be an implicit definition of "satisfaction of formulas in $\mathbb{N}$". By this we mean that we want to write up a description $\mathsf{Sat}(\mathsf{sat})$ of a binary relation $\mathsf{sat}(x, y)$ with the intended meaning that "the formula x is satisfied

[25] E.g., $H \subseteq \omega$ can be defined by $\Sigma(H) = \{\forall x(\varphi_n(x) \to H(x)) : n \in H\} \cup \{\forall x(\varphi_n(x) \to \neg H(x)) : n \notin H\}$ where $\varphi_n(x)$ is an explicit definition for $n \in \omega$.

[26] We say that a logic does not have the BDP if there are theories Th and $\Sigma(R)$ such that the language of $\Sigma(R)$ is that of Th extended with a new basic relation symbol R such that $\Sigma(R)$ is an implicit definition that does not have an equivalent explicit definition.

in $\mathbb{N}$ when the free variables of x are evaluated according to the evaluation y of variables". For this, we have to code formulas and evaluations of variables as elements of $\mathbb{N}$. Concretely, we want the following conditions (S1), (S2) to hold:

(S1) $\langle \omega, +, \star, R\rangle \models \mathsf{Sat}(\mathsf{sat})$ holds for a unique binary relation $R \subseteq \omega \times \omega$.

(S2) The meaning of the R in (S1) is "satisfaction of formulas in $\mathbb{N}$". I.e., we define a number $\ulcorner\varphi\urcorner \in \omega$ to any formula φ on the language of $\mathbb{N}$ such that for all evaluations $k : \{v_i : i \in \omega\} \to \omega$ of the variables and for all $h \in \omega$ if $\mathsf{mem}(h, i, k(v_i))$ for each free variable v_i of φ, then

$$\mathbb{N} \models \varphi[k] \quad \Leftrightarrow \quad \mathbb{N} \models \mathsf{sat}(\ulcorner\varphi\urcorner, h) \ .$$

We begin by coding formulas as numbers. We will try to code formulas as simply as we can. For simplicity, we will consider $+, \star$ to be ternary relation symbols rather than binary function symbols, and we will use the so-called Polish notation, e.g., we write $+(v_i, v_j, v_k)$ in place of $v_i + v_j = v_k$, and $\vee\varphi\psi$ in place of $(\varphi \vee \psi)$. The variables of the coded language will be v_i for $i \in \omega$.

Let us code the basic symbols $v, =, +, \star, \neg, \vee, \exists$ by the numbers 0,1,2,3,4,5,6 respectively (we could have chosen any other seven distinct numbers). In the following when we write v, it is an abbreviation for 0, etc. Below, $\langle x, y\rangle$ denotes $\mathsf{pair}(x, y)$, $\langle x, y, z\rangle$ denotes $\mathsf{pair}(x, \mathsf{pair}(y, z))$, etc. We define

$$\begin{array}{lll}
\ulcorner v_i = v_j\urcorner & \text{as} & \langle =, v, i, v, j\rangle, \\
\ulcorner +(v_i, v_j, v_k)\urcorner & \text{as} & \langle +, v, i, v, j, v, k\rangle, \\
\ulcorner \star(v_i, v_j, v_k)\urcorner & \text{as} & \langle \star, v, i, v, j, v, k\rangle, \\
\ulcorner \neg\varphi\urcorner & \text{as} & \langle \neg, \ulcorner\varphi\urcorner\rangle, \\
\ulcorner \varphi \vee \psi\urcorner & \text{as} & \langle \vee, \ulcorner\varphi\urcorner, \ulcorner\psi\urcorner\rangle, \\
\ulcorner \exists v_i\varphi\urcorner & \text{as} & \langle \exists, v, i, \ulcorner\varphi\urcorner\rangle.
\end{array}$$

For example, the code of the formula $v_0 = v_0$ is 1 because

$$\begin{aligned}
&\ulcorner v_0 = v_0 \urcorner = \\
&\langle =, v, 0, v, 0\rangle = \\
&\mathsf{pair}(=, \mathsf{pair}(v, \mathsf{pair}(0, \mathsf{pair}(v, 0)))) = \\
&\mathsf{pair}(1, \mathsf{pair}(0, \mathsf{pair}(0, \mathsf{pair}(0, 0)))) = \\
&\mathsf{pair}(1, \mathsf{pair}(0, \mathsf{pair}(0, (0+0)^2 + 0))) = \\
&\mathsf{pair}(1, \mathsf{pair}(0, \mathsf{pair}(0, 0)) = \\
&\mathsf{pair}(1, \mathsf{pair}(0, 0)) = \\
&\mathsf{pair}(1, 0) = 1.
\end{aligned}$$

We want to define a unary relation Fm with $\mathsf{Fm}(x)$ meaning that "x is the code of a formula". The natural implicit definition would start as:

$$\begin{aligned}
(*)\ \{&\mathsf{Fm}(=vivj),\ \ \mathsf{Fm}(+vivjvk),\ \ \mathsf{Fm}(\star vivjvk),\\
&\mathsf{Fm}(x) \to \mathsf{Fm}(\neg x),\ \ \mathsf{Fm}(x) \wedge \mathsf{Fm}(y) \to \mathsf{Fm}(\vee xy),\ \ \mathsf{Fm}(x) \to \mathsf{Fm}(\exists vix)\}.
\end{aligned}$$

In the above, i, j, k, x, y are variables, so, e.g., the open formula $\mathsf{Fm}(=vivj)$ in the set stands for $\forall ij\mathsf{Fm}(=vivj)$, i.e., for $\forall ij\mathsf{Fm}(10i0j)$, and $10i0j$ abbreviates $\langle 1, 0, i, 0, j\rangle$. Now, (*) above is not a definition in $\mathbb{N}$, because more than one sets satisfy it in $\mathbb{N}$ (one is $\{\ulcorner\varphi\urcorner : \varphi \in \mathcal{L}(\mathbb{N})\}$ while another is ω itself). The usual condition "and only these are formulas" is missing from (*). Because of this, we define $\mathsf{Fm}(x)$ in a different way, namely we define it explicitly as

$$\mathsf{Fm}(x) \quad \leftrightarrow \quad \exists si(s_i = x \wedge \mathsf{deriv}(s, i))$$

where $\mathsf{deriv}(s, i)$ denotes the formula below. For easier readability, we will use conventions connected with defined constants. We denote by s_i the unique number for which $\mathsf{mem}(s, i, s_i)$ holds when s is a pair, and then, e.g., $s_i = \langle\neg, s_k\rangle$ abbreviates the formula $\exists zw(\mathsf{mem}(s, i, z) \wedge \mathsf{mem}(s, k, w) \wedge z = \langle\neg, w\rangle)$.

$$\begin{aligned}
\mathsf{deriv}(s, i) \leftrightarrow \forall j \le i\ [\ \ &\exists kl\ s_j = \langle =, v, k, v, l\rangle\ \ \vee\\
&\exists klm\ s_j = \langle +, v, k, v, l, v, m\rangle\ \ \vee\\
&\exists klm\ s_j = \langle \star, v, k, v, l, v, m\rangle\ \ \vee\\
&\exists k < j\ s_j = \langle\neg, s_k\rangle\ \ \vee\\
&\exists k < j \exists l < j\ s_j = \langle\vee, s_k, s_l\rangle\ \ \vee\\
&\exists k < j \exists n\ s_j = \langle\exists, v, n, s_k\rangle\ \ ]\,.
\end{aligned}$$

We define the formula $\mathsf{upvar}(x, n)$ to mean that in the formula coded by x only variables v_k with $k \leq n$ occur.

$$\begin{aligned}\mathsf{upvar}(x,n) \leftrightarrow \exists si(\quad & \mathsf{deriv}(s,i) \quad \wedge \quad s_i = x \quad \wedge \quad \forall j \leq i \, [\\ & s_j = \langle =, v, k, v, l\rangle \rightarrow (k \leq n \wedge l \leq n) \quad \wedge \\ & s_j = \langle +, v, k, v, l, v, m\rangle \rightarrow (k \leq n \wedge l \leq n \wedge m \leq n) \wedge \\ & s_j = \langle \star, v, k, v, l, v, m\rangle \rightarrow (k \leq n \wedge l \leq n \wedge m \leq n) \wedge \\ & s_j = \langle \exists, v, l, s_k\rangle \rightarrow l \leq n \,] \,) \, .\end{aligned}$$

We are ready to define $\mathsf{Sat}(\mathsf{sat})$. For easier readability, we define $\mathsf{eval}(y) \leftrightarrow \exists zw \, y = \mathsf{pair}(z, w)$. The reason for the name is that each pair codes an infinite sequence via the convention mem. To make the formulas more readable, we leave out the outermost universal quantifiers of formulas, e.g., we write $\mathsf{sat}(x, y) \rightarrow \mathsf{Fm}(x)$ in place of $\forall xy[\mathsf{sat}(x, y) \rightarrow \mathsf{Fm}(x)]$.

$$\begin{aligned}\mathsf{Sat}(\mathsf{sat}) = \{ \quad & \mathsf{sat}(x,y) \rightarrow \mathsf{Fm}(x), \quad \mathsf{sat}(x,y,) \rightarrow \mathsf{eval}(y), \\ & \mathsf{sat}(\langle =, v, k, v, l\rangle, y) \;\leftrightarrow\; y_k = y_l, \\ & \mathsf{sat}(\langle +, v, k, v, l, v, m\rangle, y) \;\leftrightarrow\; y_k + y_l = y_m, \\ & \mathsf{sat}(\langle \star, v, k, v, l, v, m\rangle, y) \;\leftrightarrow\; y_k \star y_l = y_m, \\ & \mathsf{sat}(\langle \neg, x\rangle, y) \;\leftrightarrow\; (\neg\mathsf{sat}(x,y) \wedge \mathsf{Fm}(x) \wedge \mathsf{eval}(y)), \\ & \mathsf{sat}(\langle \vee, x, z\rangle, y) \;\leftrightarrow\; (\mathsf{sat}(x,y) \vee \mathsf{sat}(z,y)), \\ & \mathsf{sat}(\langle \exists, v, l, x\rangle, y) \;\leftrightarrow \\ & \exists y' n(\mathsf{upvar}(x,n) \wedge \forall i \leq n, i \neq l \; y'_i = y_i \wedge \mathsf{sat}(x,y')) \quad \} \, .\end{aligned}$$

We want to show that $(S1), (S2)$ hold. To this end, first we show that

$$\mathsf{Fm}(x) \;\text{ iff }\; x = \ulcorner\varphi\urcorner \;\text{ for some formula }\; \varphi \, .$$

We can show this by induction of the lengths of the derivations for x. I.e., we prove by induction on i that $\forall s \, \mathsf{deriv}(s, i) \rightarrow \forall j \leq i \, \exists \varphi_j \; s_j = \ulcorner\varphi_j\urcorner$. Note that the above formula is on the metalanguage, and not on the language of $\mathbb{N}$. Then we show that

$$\text{the formula } \varphi \text{ for which } x = \ulcorner\varphi\urcorner \text{ is unique} \, .$$

Proving $\forall\psi \; (\ulcorner\varphi\urcorner = \ulcorner\psi\urcorner \rightarrow \varphi = \psi)$ by induction on φ suffices for this. Then by a similar induction, we prove

$$\mathsf{upvar}(\ulcorner\varphi\urcorner, n) \;\text{ iff the variables occurring in } \varphi \text{ are among } \; \{v_i : i \leq n\} \, .$$

When y is a pair, let $\overline{y}$ denote the evaluation that assigns y_j to each variable v_j. Then we prove by induction on the length of the derivation of φ that

$$\mathsf{Sat}(\mathsf{sat}) \wedge \mathsf{sat}(x, y) \to \mathbb{N} \models \varphi[\overline{y}] \quad \text{whenever } x = \ulcorner\varphi\urcorner .$$

Finally, we show

$$\mathsf{Sat}(\mathsf{sat}) \wedge \mathbb{N} \models \varphi[k] \to \mathsf{sat}(\ulcorner\varphi\urcorner, h) \text{ whenever } \mathsf{upvar}(\ulcorner\varphi\urcorner, n) \wedge \forall i \leq n\ h_i = k(v_i) .$$

These prove both (S1) and (S2).

Having shown that sat really expresses satisfiability of formulas in $\mathbb{N}$, we turn to proving that it cannot be explicitly defined in $\mathbb{N}$. Assume, contrary, that it can be defined by the concrete formula $T(x, y)$, i.e.,

(S3) $\mathbb{N} \models \mathsf{Sat}(\mathsf{sat}) \to \forall xy[T(x, y) \leftrightarrow \mathsf{sat}(x, y)]$.

We may assume that the variables x, y are v_0, v_1. Define

$$F(x) \quad \leftrightarrow \quad \forall y(y_0 = x \to \neg T(x, y)) .$$

(Note that $y_0 = x$ above abbreviates $\mathsf{mem}(y, 0, x)$.) One can interpret $F(x)$ as expressing that the formula x is not satisfied at itself. Let

$$t = \ulcorner F(x)\urcorner .$$

We want to know whether $F(x)$ is true in $\mathbb{N}$ when x is evaluated for this t. Let $\langle t\rangle$ denote any pair (i.e., evaluation) h for which $h_0 = t$. Below, we will write $[x \to t]$ for any evaluation k for which $k(x) = t$ and similarly for $[y \to \langle t\rangle]$.

$\mathbb{N} \models F(x)[x \to t]$	iff	by definition of $\models$
$\mathbb{N} \models F(x)[k]$, where $k(x) = t$	iff	by (S2)
$\mathbb{N} \models \mathsf{sat}(\ulcorner F(x)\urcorner, \langle t\rangle)$	iff	by (S3) and $t = \ulcorner F(x)\urcorner$
$\mathbb{N} \models T(x, y)[x \to t, y \to \langle t\rangle]$	iff	by definition of F(x))
$\mathbb{N} \models \neg F(x)[x \to t]$.		

We arrived at a contradiction by assuming the existence of an explicit definition for $\mathsf{Sat}(\mathsf{sat})$. We infer that there is no explicit definition for $\mathsf{Sat}(\mathsf{sat})$ that works in $\mathbb{N}$.

2.4 Definitions in Second-order Logic SOL

Second-order logic SOL is FOL enriched with variable symbols for n-place relations, for all $n \in \omega$. We will also care for *weak second-order logic* (wSOL), where only variables ranging over finite subsets are added. We note that when X is a variable for a unary relation and x is an individual variable, $X(x)$ denotes that "x has property X". A unary relation is just a subset, so often we will say that X ranges over subsets of the universe and we can then also write $x \in X$ for $X(x)$. Thus wSOL is a fragment of SOL.

The fact that $\mathsf{Sat}(\mathsf{sat})$ is not equivalent in $\mathbb{N}$ to any explicit definition does not contradict Theorem 1.1 because $\mathbb{N}$ is not a FOL-axiomatizable class of models. SOL, however, is a stronger logic than FOL, and by SOL-formulas $\mathbb{N}$ can be axiomatized. E.g., it can be axiomatized by the conjunction of the first four statements in the definition of PA (Definition 2.2) with the following SOL-formula

$$\forall X \left[(X(0) \wedge \forall x (X(x) \to X(\mathsf{suc}(x))) \to \forall x \, X(x) \right].$$

(This last formula is called *second-order induction axiom* and the last schema in the definition of PA is the scheme we obtain from this when we restrict the variable X to range over subsets defined by FOL-formulas φ.)

Does this prove that SOL does not have BDP? We have a SOL-theory and an implicit definition $\mathsf{Sat}(\mathsf{sat})$ that works in each model of this theory and which is not equivalent to any explicit definition. Well, this implicit definition is not equivalent to any explicit FOL definition. Just because SOL is stronger than FOL, it may have an explicit SOL definition. The following theorem says that it indeed does.

Theorem 2.5 *(SOL has BDP for finite implicit definitions) Let $\Sigma(R)$ be a finite implicit definition in SOL. Then the following SOL-formula is an explicit definition for R:*

$$R(\overline{x}) \leftrightarrow \exists X (\bigwedge \Sigma(X) \wedge X(\overline{x})). \qquad \square$$

We note that SOL does not have BDP, because there are (in $\mathbb{N}$) infinite implicit definitions that are not equivalent to any SOL-explicit definitions (see the cardinality argument at the beginning of section 2.3).

The opening idea of this section can be realized, however. Weak SOL is between FOL and SOL in expressive power. We are going to show that $\mathbb{N}$

still can be axiomatized in wSOL, but there is no wSOL explicit definition equivalent to (a slightly modified version of) $\mathsf{Sat}(\mathsf{sat})$.

Axiomatization of $\mathbb{N}$ in wSOL: The following formula in conjunction with the first four statements in the definition of PA axiomatizes $\mathbb{N}$. Keep in mind that X is a second-order variable ranging over *finite* sets only.

$$\forall x \exists X(X(0) \wedge \forall y[(X(y) \wedge y \neq x) \rightarrow X(\mathsf{suc}(y))]).$$

Next we show that there is no wSOL-formula which would be an explicit definition for $\mathsf{Sat}(\mathsf{sat})$. In order to use the "self-referential" idea of why $\mathsf{Sat}(\mathsf{sat})$ cannot be made explicit in FOL, we have to modify it to express satisfiability of all wSOL-formulas in place of all FOL-formulas. Below we include this modified version of $\mathsf{Sat}(\mathsf{sat})$.

In wSOL we have set-variables V_i (ranging over finite sets) also besides the variables v_i (ranging over elements), we have quantifiers $\exists V_i$, and we have primitive formulas of form $V_i(v_j)$ for $i, j \in \omega$. To code formulas of wSOL as numbers, let us assign the number 7 to V, and let us add the following two lines to the definition of $\ulcorner\varphi\urcorner$ for FOL-formulas to get the codes of all wSOL-formulas:

$$\begin{array}{lll} \ulcorner V_i(v_j)\urcorner & \text{as} & \langle V, i, v, j\rangle, \\ \ulcorner \exists V_i \varphi\urcorner & \text{as} & \langle \exists, V, i, \ulcorner\varphi\urcorner\rangle. \end{array}$$

Let us obtain the formula deriv^w from deriv by adding two lines to it, namely

$$\begin{array}{ll} \exists k \ell & s_j = \langle V, k, v, \ell\rangle \ \vee \\ \exists k < j \exists n & s_j = \langle \exists, V, n, s_k\rangle \end{array}$$

Then let Fm^w denote the formula we get from Fm by replacing deriv with deriv^w in it. When x is the code of a wSOL-formula, let $\mathsf{upvar}^w(x, n)$ express that in the formula coded by x only variables v_k with $k \leq n$ and set-variables V_k with $k \leq n$ occur. This can be expressed by changing deriv to deriv^w in the definition of upvar and adding the following two lines to it:

$$\begin{array}{l} s_j = \langle V, k, v, l\rangle \rightarrow (k \leq n \wedge l \leq n) \wedge \\ s_j = \langle \exists, V, l, s_k\rangle \rightarrow l \leq n \end{array}$$

Having coded wSOL formulas, let us code evaluations. Evaluations of variables in wSOL assign elements to the individual variables v_i and finite subsets to the set-variables V_i. We will code finite subsets by ranges of

sequences.[27] The reason is that each sequence coded by a pair is periodic, hence its range is finite.[28] Therefore it will be convenient for us to code satisfiability of wSOL-formulas as a three-place relation $\mathsf{wsat}(x, h, s)$ where x is the code of a wSOL formula, h is a pair (thus coding an infinite sequence by the convention embodied in mem) and s is an infinite sequence whose members that correspond to variables occurring in the formula x are all sequences:

$$\mathsf{eval}^w(s,x) \leftrightarrow \exists yz\, s = \mathsf{pair}(y,z) \wedge \exists n[\mathsf{upvar}^w(x,n) \wedge \forall i \leq n \exists y'z'\, s_i = \mathsf{pair}(y',z')].$$

Let us introduce the notation

$$y \in \widetilde{s}_k \leftrightarrow \exists i\, y = (s_k)_i.$$

Below, the variables $x, y, s, y', s', z, k, l, m, n, i$ denote the individual variables $v_0, ..., v_{11}$.

$$\begin{aligned}
\mathsf{Sat}^w(\mathsf{wsat}) = \{\ & \mathsf{wsat}(x,y,s) \rightarrow (\mathsf{Fm}^w(x) \wedge \mathsf{eval}(y) \wedge \mathsf{eval}^w(s,x)),\\
& \mathsf{wsat}(\langle V,k,v,l\rangle, y, s) \ \leftrightarrow\ y_l \in \widetilde{s}_k,\\
& \mathsf{wsat}(\langle =,v,k,v,l\rangle, y, s) \ \leftrightarrow\ y_k = y_l,\\
& \mathsf{wsat}(\langle +,v,k,v,l,v,m\rangle, y, s) \ \leftrightarrow\ y_k + y_l = y_m,\\
& \mathsf{wsat}(\langle \star,v,k,v,l,v,m\rangle, y, s) \ \leftrightarrow\ y_k \star y_l = y_m,\\
& \mathsf{wsat}(\langle \neg,x\rangle, y, s) \ \leftrightarrow\\
& \quad (\neg\mathsf{wsat}(x,y,s) \wedge \mathsf{Fm}^w(x) \wedge \mathsf{eval}(y) \wedge \mathsf{eval}^w(s,x)),\\
& \mathsf{wsat}(\langle \vee,x,z\rangle, y, s) \ \leftrightarrow\ (\mathsf{wsat}(x,y,s) \vee \mathsf{wsat}(z,y,s)),\\
& \mathsf{wsat}(\langle \exists,v,l,x\rangle, y, s) \ \leftrightarrow \exists y'n\\
& \quad (\mathsf{upvar}^w(x,n) \wedge \forall i \leq n, i \neq l\ y'_i = y_i \wedge \mathsf{wsat}(x,y',Ss)),\\
& \mathsf{wsat}(\langle \exists,V,l,x\rangle, y, s) \ \leftrightarrow \exists s'n\\
& \quad (\mathsf{upvar}^w(x,n) \wedge \forall i \leq n, i \neq l\ \widetilde{s}'_i = \widetilde{s}_i \wedge \mathsf{wsat}(x,y,s'))\ \ \}\,.
\end{aligned}$$

As before, we can show that Sat^w is an implicit definition in $\mathbb{N}$. Assume $T(x,y,s)$ is a SOL-formula which defines $\mathsf{wsat}(x,y,s)$ explicitly in $\mathbb{N}$, we define $F(x)$ as $\forall ys(y_0 = x \rightarrow \neg T(x,y,s))$, and with this formula we can repeat the previous argument to arrive at a contradiction. Thus, Sat^w is an

[27] This is the step which does not go through for SOL. In the modified version of $\mathsf{Sat}(\mathsf{sat})$ for all SOL-formulas, we need as argument of sat an evaluation of variables to arbitrary sets, and arbitrary subsets of $\mathbb{N}$ cannot be "coded" by elements of $\mathbb{N}$, while finite sets can.

[28] It is not necessary to rely on this fact, we also could code finite sequences as ranges of initial segments of sequences.

implicit definition in the wSOL-theory of $\mathbb{N}$ which is not equivalent to any explicit wSOL-definition. This shows that wSOL does not have the Beth Definability Property.[29]

2.5 Complexity of explicit definitions

In this part we show that an implicit definition can be much simpler than the equivalent explicit definition in the sense that we measure simplicity by the number of variables used in the formulas.

Let n be any finite number (i.e., $n \in \omega$). The n-variable fragment $\mathcal{L}_n$ of a FOL language $\mathcal{L}$ is the set of all formulas in $\mathcal{L}$ which use the first n variables only (free or bound). To make this meaningful, we can assume that $\mathcal{L}$ uses the variables $v_0, v_1, ...$ while $\mathcal{L}_n$ uses only the variables $v_0, v_1, ..., v_{n-1}$. In finite variable fragments we do not allow function or constant symbols. Here is a definition of the formulas of $\mathcal{L}_n$:

$R(v_{i_1}, ..., v_{i_k})$ is a formula of $\mathcal{L}_n$ if R is a k-place relation symbol and $i_1, ..., i_k < n$.

$v_i = v_j$ is a formula of $\mathcal{L}_n$ if $i, j < n$.

$\neg\varphi$, $\varphi \wedge \psi$, $\exists v_i \varphi$ are formulas of $\mathcal{L}_n$ whenever φ, ψ are formulas of $\mathcal{L}_n$ and $i < n$.

The above are all the formulas of $\mathcal{L}_n$. Models, satisfiability of formulas under evaluations of the variables, validity in $\mathcal{L}_n$ are the same as in FOL. $\mathcal{L}_n$ does not have even the weak Beth Definability Property whenever $n \geq 3$.:

Theorem 2.6 *(No weak Beth Property for $\mathcal{L}_n$.) Let $n \geq 3$. There are a theory Th in the language of an n-place relation symbol R together with a binary relation symbol s and a description $\Sigma(D)$ for a unary relation D such that $\Sigma(D)$ is an implicit definition of D in Th but there is no explicit definition for D in Th, i.e., for each n-variable formula φ in the language of Th we have*

$$\mathsf{Th} \cup \Sigma(D) \not\models \forall v_0[D(v_0) \leftrightarrow \varphi] \ .$$

[29] Basically this proof for wSOL not having the (weak) Beth Definability Property is given in [19, item 7.2 on p.102] and in [3, pp.74-75]. Another nice proof, due to Tarski, is given in [19, below item 7.2 on p.102].

Theorem 1.1 implies that there is a FOL-formula for Th and $\Sigma(D)$ as in Thm.2.6 which explicitly defines $D(v_0)$. The above theorem then implies that this explicit definition has to use more than n variables. Thus, both the theory and the implicit definition use only n variables, but any equivalent explicit definition has to use more than n variables. In our case, there is an explicit definition that uses $n+1$ variables. However, for each $n \in \omega$, there is an implicit definition that uses only 3 variables and each explicit definition equivalent to it has to use more than n variables, see [15].

Proof. We write out the proof for $n = 3$. Generalizing this proof to all $n \geq 3$ will be easy. We will often write x, y, z for v_0, v_1, v_2 and we will write simply R for $R(x, y, z)$. Let

$$U_0(x) \leftrightarrow \exists yzR, \qquad U_1(y) \leftrightarrow \exists xzR, \qquad U_2(z) \leftrightarrow \exists xyR.$$

These formulas express the domain of R, i.e., the first projection of R, and the second and third projections of R. We will include formulas into Th that express that U_0, U_1, U_2 are sets of cardinalities $3, 2, 2$ respectively, and they form a partition of the universe. We will formulate these properties with 3 variables after describing the main part of the construction. Let

$$T \leftrightarrow U_0(x) \wedge U_1(y) \wedge U_2(z),$$
$$\mathsf{big}(R) \leftrightarrow \bigwedge\{\exists v_i R \leftrightarrow \exists v_i(T \wedge \neg R) : i = 0, 1, 2\}.$$

In the above, T is the "rectangular hull" of R, and $\mathsf{big}(R)$ expresses that R cuts this T into two parts each of which is sensitive in the sense that as soon as we quantify over them, the information on how R cuts T into two parts disappears. (Note that $\mathsf{big}(R)$ implies that $\exists v_i R \leftrightarrow \exists v_i T \leftrightarrow \exists v_i(T \wedge \neg R)$.) Assume that $|U_0| = 3, |U_1| = 2, |U_2| = 2$ and $\mathsf{partition}(U_0, U_1, U_2)$ are formulas in $\mathcal{L}_3$ that express the associated meanings. Then

$$\mathsf{Th} = \{|U_0| = 3, |U_1| = 2, |U_2| = 2, \quad \mathsf{partition}(U_0, U_1, U_2), \quad \mathsf{big}(R)\}.$$

We will show that Th has exactly one model, up to isomorphisms. But before doing that, lets turn to expressing the properties we promised about the U_i's with using three variables.

We will use *Tarski's way of substituting* one variable for the other. Let

$$U_1(x) \leftrightarrow \exists y(x = y \wedge U_1(y)), \qquad U_2(x) \leftrightarrow \exists z(x = z \wedge U_2(z)).$$

We now can express that U_0, U_1, U_2 form a partition of the universe:

$$\forall x(U_0(x) \vee U_1(x) \vee U_2(x)), \qquad \forall x(U_i(x) \to \neg U_j(x)) \quad \text{for} \quad i \neq j, \quad i, j < 3.$$

For expressing the sizes of the sets U_i we will use the abbreviations

$$U_1(z) \leftrightarrow \exists z(z = y \wedge U_1(y)), \qquad U_2(y) \leftrightarrow \exists z(y = z \wedge U_2(z)).$$

Now, for $i = 1, 2$ we define the formulas

$$\begin{array}{l} |U_i| \leq 2 \leftrightarrow \neg\exists xyz(x \neq y \wedge x \neq y \wedge y \neq z \wedge U_i(x) \wedge U_i(y) \wedge U_i(z)), \\ |U_i| \geq 2 \leftrightarrow \exists xy(x \neq y \wedge U_i(x) \wedge U_i(y)), \\ |U_i| = 2 \leftrightarrow |U_i| \geq 2 \wedge |U_i| \leq 2. \end{array}$$

It remains to express that U_0 has exactly three elements. In $\mathcal{L}_n$ with $n \geq 4$ we can express $|U_0| = 3$ similarly to the above, but in $\mathcal{L}_3$ we have to use another tool. For expressing in $\mathcal{L}_3$ that U_0 has exactly 3 elements, we will use the binary relation s. (This is the sole use of s in Th.) We are going to express that s is a cycle of order 3 on U_0. The following formulas express that s is a function on U_0 without a fixed-point:

$$\forall x\exists y\, s(x, y), \quad s(x, y) \wedge s(x, z) \to y = z, \quad s(x, y) \to (U_0(x) \wedge U_0(y) \wedge x \neq y).$$

The following formula expresses that U_0 consists of exactly one 3-cycle of s:

$$s(y, x) \leftrightarrow \exists z(s(x, z) \wedge s(z, y)), \quad s(x, y) \vee s(y, x) \vee x = y.$$

In the above, we used Tarski-style substitution of variables without mentioning (e.g., $U_0(y)$) and we omitted universal quantifiers in front of formulas (e.g., we wrote $s(x, y) \wedge s(x, z) \to y = z$ in place of $\forall xy(s(x, y) \wedge s(x, z) \to y = z)$).

We turn to showing that Th has exactly one model up to isomorphism. Let $\mathfrak{M} = \langle M, R, s\rangle \models \mathsf{Th}$. Let U_i, T be defined as above. Then M is the disjoint union of the U_i's, and the sizes of the U_i's for $i = 0, 1, 2$ are 3,2,2 respectively. (So M has 7 elements.) Let $U_1 = \{b_0, b_1\}$, let c, d be the two elements of U_2 and let

$$X = \{u \in U_0 : \langle u, b_0, c\rangle \in R\}.$$

By $\mathfrak{M} \models \mathsf{big}(R)$ we have that $\langle u, b_0, d\rangle \notin R$ if $u \in X$ and $\langle u, b_0, d\rangle \in R$ if $u \in U_0 - X$. Hence

$$U_0 - X = \{u \in U_0 : \langle u, b_0, d\rangle \in R\}.$$

Also, by $\mathfrak{M} \models \mathsf{big}(R)$, X has one, or X has two elements. If $|X| = 1$ then let's use the notation $c_0 = c, c_1 = d$, and if $|X| = 2$ then let $c_0 = d, c_1 = c$. Let us name the elements of U_0 as a_0, a_1, a_2 such that $X = \{a_0\}$ if $|X| = 1$, $X = \{a_1, a_2\}$ if $|X| = 2$ and $s = \{\langle a_i, a_j\rangle : j = i + 1(mod 3) \text{ and } i, j \leq 3\}$. This can be done by $\mathfrak{M} \models \mathsf{Th}$. The setting so far determines R by $\mathfrak{M} \models \mathsf{big}(R)$, as follows. For all $i, j, k \leq 2$ we have $\langle a_i, b_j, c_k\rangle \in R$ if and only if $\langle a_i, b_{j+1(mod 2)}, c_k\rangle \in T - R$ if and only if $\langle a_i, b_j, c_{k+1(mod 2)}\rangle \in T - R$. This is so by $\mathfrak{M} \models \mathsf{big}(R)$ and by $|U_i| = 2$ for $i = 1, 2$. From this we have that

$$\begin{array}{lll} R = & \{\langle u, b_i, c_j\rangle : u = a_0 & \text{and } i + j = 0(mod 2)\} \cup \\ & \{\langle u, b_i, c_j\rangle : u = a_1 \vee u = a_2 & \text{and } i + j = 1(mod 2)\}\ . \end{array}$$

We have seen that all models of Th are isomorphic to each other. By using the above ideas, one can also see that there is no automorphism[30] of $\mathfrak{M}$ that would move $\{a_0\}$.

We are ready to formulate our implicit definition $\Sigma(D)$. It will single out $\{a_0\}$ in the above notation. We will write D in place of $D(x)$.

$$\begin{array}{lll} \Sigma(D) = \{ & T \wedge \neg D \wedge R & \rightarrow \quad \forall x(T \wedge \neg D \rightarrow R)), \\ & T \wedge \neg D \wedge \neg R & \rightarrow \quad \forall x(T \wedge \neg D \rightarrow \neg R)), \\ & D \rightarrow U_0(x), \quad |D| = 1 & \}. \end{array}$$

Then in each model of Th there is exactly one unary relation D for which $\Sigma(D)$ holds, namely D has to be the unary relation $\{a_0\} \subseteq U_0$. Thus $\Sigma(D)$ is a strong implicit definition of D in Th.

We show that Σ cannot be made explicit in $\mathcal{L}_3$, i.e., there is no 3-variable formula φ in the language of Th for which $\mathsf{Th} \cup \Sigma(D) \models D \leftrightarrow \varphi$. Our plan is to list all the $\mathcal{L}_3$-definable relations in the above model and observe that $\{a_0\}$, the relation Σ defines, is not among them. For any $\varphi \in \mathcal{L}_3$ define

$$\mathsf{mn}(\varphi) = \{\langle a, b, c\rangle : \mathfrak{M} \models \varphi[a, b, c]\}.$$

[30] Automorphism of $\mathfrak{M}$ means isomorphism between $\mathfrak{M}$ and $\mathfrak{M}$.

In the above, $\mathfrak{M} \models \varphi[a,b,c]$ denotes that the formula φ is true in $\mathfrak{M}$ when the variables v_0, v_1, v_2 are evaluated to a, b, c respectively, and mn abbreviates "*meaning*". Let

$$A = \{\mathsf{mn}(\varphi) : \varphi \in \mathcal{L}_3\}.$$

Clearly, A is closed under the set Boolean operations because

$$\begin{aligned} &\mathsf{mn}(\varphi \wedge \psi) = \mathsf{mn}(\varphi) \cap \mathsf{mn}(\psi), \\ &\mathsf{mn}(\neg\varphi) = M^3 - \mathsf{mn}(\varphi), \end{aligned}$$

and so A is closed under intersection and complementation w.r.t. M^3. Since M is finite, this implies that A is atomic[31] and the elements of A are exactly the unions of some atoms.

We will list all the atoms of A. It is easy to see that the elements $U_i \times U_j \times U_k$ for $i,j,k \leq 2$ are all in A and they form a partition of M^3. To list the atoms of A, we will list the atoms below each $U_i \times U_j \times U_k$ by specifying a partition of each. Let $i,j,k \leq 2$. Let's abbreviate the sequence $\langle i,j,k\rangle$ by ijk.

Assume i,j,k are all distinct, i.e., $|\{i,j,k\}| = 3$. We define

$$\begin{aligned} &X(ijk,r) = \{\langle u_i,u_j,u_k\rangle : \langle u_0,u_1,u_2\rangle \in R\}, \\ &X(ijk,-r) = \{\langle u_i,u_j,u_k\rangle \in U_i \times U_j \times U_k : \langle u_0,u_1,u_2\rangle \notin R\}. \end{aligned}$$

We note that

$$X(012,r) = R, \quad \text{and} \quad X(012,-r) = T - R.$$

For i,j,k a permutation of $0,1,2$, $X(ijk,r)$ and $X(ijk,-r)$ are the correspondingly permuted versions of R and $T-R$. In particular,

$$\mathsf{mn}(R(v_i,v_j,v_k)) = X(ijk,r).$$

Assume now that ijk is not repetition-free, i.e., $|\{i,j,k\}| < 3$. In these cases the blocks of the partition of $U_i \times U_j \times U_k$ will be put together from partitions of $U_m \times U_n$ $(m,n<3)$. Recall that $s = \{\langle a_0,a_1\rangle, \langle a_1,a_2\rangle, \langle a_2,a_0\rangle\}$. We define

$$\begin{aligned} &\overline{s} = \{\langle a,b\rangle : \langle b,a\rangle \in s\}, \\ &\mathsf{id}_i = \{\langle a,a\rangle : a \in U_i\}, \\ &\mathsf{di}_i = \{\langle a,b\rangle : a \neq b, \quad a,b \in U_i\}. \end{aligned}$$

[31] An atom in a Boolean algebra is a minimal non-zero element, and a Boolean algebra is atomic if below each non-zero element there is an atom.

Above, $\mathsf{id}_i, \mathsf{di}_i$ abbreviate "*identity* on U_i", and "*diversity* on U_i", respectively, and $\overline{s}$ is the inverse of s. Since s is a cycle on the three-element set U_0, its inverse $\overline{s}$ is its complement in the diversity element of U_0, so $\{s, \overline{s}, \mathsf{id}_0\}$ is a partition of $U_0 \times U_0$. Since U_1 is a two-element set, $\{\mathsf{di}_1, \mathsf{id}_1\}$ is a partition of $U_1 \times U_1$, and the the same holds for U_2. We are ready to define the "binary partitions" as follows

$$\begin{array}{l} \mathsf{Rel}_{00} = \{s, \overline{s}, \mathsf{id}_0\}, \quad \mathsf{Rel}_{11} = \{\mathsf{di}_1, \mathsf{id}_1\}, \quad \mathsf{Rel}_{22} = \{\mathsf{di}_2, \mathsf{id}_2\}, \\ \mathsf{Rel}_{ij} = \{U_i \times U_j\} \ \text{ for } \ i \neq j. \end{array}$$

We say that "e is a *good choice* for ijk", in symbols $\mathsf{choice}(e, ijk)$, if

$$\begin{array}{ll} e \in \{r, -r\} & \text{when } \ |\{i,j,k\}| = 3, \quad \text{otherwise} \\ \\ e = \langle e_{01}, e_{12}, e_{02} \rangle & \text{where } \ e_{01} \in \mathsf{Rel}_{ij}, \quad e_{12} \in \mathsf{Rel}_{jk}, \quad e_{02} \in \mathsf{Rel}_{ik}, \\ & \text{and} \quad e_{02} = e_{01} \circ e_{12} \quad \text{if } \ 0 = i = j = k. \end{array}$$

For example, $e = \langle s, s, \overline{s} \rangle$ is a good choice for 000, $e = \langle \mathsf{di}_1, \mathsf{id}_1, \mathsf{di}_1 \rangle$ is a good choice for 111, $e = \langle \overline{s}, U_0 \times U_1, U_0 \times U_1 \rangle$ is a good choice for 001 and $e = -r$ is a good choice for 012. These choices will represent the elements

$$\begin{array}{l} \{z \in U_0 \times U_0 \times U_0 : \langle z_0, z_1 \rangle, \langle z_1, z_2 \rangle \in s\}, \\ \{z \in U_1 \times U_1 \times U_1 : z_0 \neq z_1 = z_2\}, \\ \{z \in U_0 \times U_0 \times U_1 : \langle z_1, z_0 \rangle \in s\}, \\ \text{etc.} \end{array}$$

When e is a good choice for ijk and $|\{i,j,k\}| < 3$ we define

$$X(ijk, e) = \{\langle a, b, c \rangle \in U_i \times U_j \times U_k : \langle a, b \rangle \in e_{01}, \ \langle b, c \rangle \in e_{12}, \ \langle a, c \rangle \in e_{02}\},$$

$$\begin{array}{l} B = \{X(ijk, e) : i, j, k \leq 2, \mathsf{choice}(e, ijk)\}, \\ C = \{\bigcup Y : Y \subseteq B\}. \end{array}$$

We want to prove that $A = C$. We show $A \subseteq C$ by showing $\mathsf{mn}(\varphi) \in C$ for all $\varphi \in \mathcal{L}_3$, by induction on φ. Atomic formulas:

$$\begin{array}{l} \mathsf{mn}(R(v_i, v_j, v_k)) = X(ijk, r) \quad \text{when } \ |\{i,j,k\}| = 3, \\ \mathsf{mn}(R(v_i, v_j, v_k)) = \emptyset \quad \text{otherwise}, \\ \mathsf{mn}(s(v_i, v_j)) = \bigcup\{X(n_1 n_2 n_3, e) : n_i = n_j = 0, e_{n_i n_j} = s\}, \\ \\ \mathsf{mn}(v_i = v_j) = \bigcup\{X(n_1 n_2 n_3, e) : n_i = n_j, e_{n_i n_j} \in \{\mathsf{id}_0, \mathsf{id}_1, \mathsf{id}_2\}\}. \end{array}$$

Clearly, $M^3 \in C$, and C is closed by complementation and intersection because B is finite and its elements are pairwise disjoint. Thus,

$$\mathsf{mn}(\neg\varphi) \in C, \quad \mathsf{mn}(\varphi \wedge \psi) \in C \qquad \text{whenever} \quad \mathsf{mn}(\varphi), \mathsf{mn}(\psi) \in C.$$

To deal with the existential quantifiers, we define for arbitrary $X \subseteq M^3$

$$\begin{aligned}
\mathsf{C}_0 X &= \{\langle a, b, c\rangle \in M^3 : \langle a', b, c\rangle \in X \ \text{ for some } a'\},\\
\mathsf{C}_1 X &= \{\langle a, b, c\rangle \in M^3 : \langle a, b', c\rangle \in X \ \text{ for some } b'\},\\
\mathsf{C}_2 X &= \{\langle a, b, c\rangle \in M^3 : \langle a, b, c'\rangle \in X \ \text{ for some } c'\}.
\end{aligned}$$

Then we have, by the definition of the meaning of the existential quantifiers, that for all $i \leq 2$

$$\mathsf{mn}(\exists v_i \varphi) = \mathsf{C}_i \mathsf{mn}(\varphi).$$

Thus, to show that

$$\mathsf{mn}(\exists v_i \varphi) \in C \quad \text{whenever} \quad \mathsf{mn}(\varphi) \in C$$

it is enough to show that C is closed under C_i, i.e., $\mathsf{C}_i X \in C$ whenever $X \in C$ (and $i \leq 2$). Since C_i is additive, i.e., $\mathsf{C}_i(X \cup Y) = \mathsf{C}_i(X) \cup \mathsf{C}_i(Y)$, it is enough to show that

$$\mathsf{C}_m X(ijk, e) \in C \quad \text{for all} \quad i, j, k, m \leq 2, \quad \text{and good choice } e \text{ for } ijk.$$

Assume i, j, k are distinct and $e \in \{r, -r\}$. Then by $\mathfrak{M} \models \mathsf{big}(R)$

$$\begin{aligned}
\mathsf{C}_0 X(ijk, e) &= M \times U_j \times U_k,\\
\mathsf{C}_1 X(ijk, e) &= U_i \times M \times U_k,\\
\mathsf{C}_2 X(ijk, e) &= U_i \times U_j \times M.
\end{aligned}$$

When i, j, k are not all distinct

$$\begin{aligned}
\mathsf{C}_0 X(ijk, e) &= M \times e_{12} = \{\langle a, b, c\rangle : \langle b, c\rangle \in e_{12}\} =\\
&\quad \bigcup\{X(mjk, e') : m \leq 2, e'_{12} = e_{12}\},\\
\mathsf{C}_1 X(ijk, e) &= \{\langle a, b, c\rangle : \langle a, c\rangle \in e_{02}\} =\\
&\quad \bigcup\{X(imk, e') : m \leq 2, e'_{02} = e_{02}\},\\
\mathsf{C}_2 X(ijk, e) &= \bigcup\{X(ijm, e') : m \leq 2, e'_{01} = e_{01}\}.
\end{aligned}$$

To show that $C \subseteq A$ we have to check that each $X(ijk, e)$ is the meaning of a formula $\varphi \in \mathcal{L}_3$ in $\mathfrak{M}$. We already did this for $X(ijk, r)$, i, jk distinct. For $ijk = 000$ and $e = \langle s, s, \overline{s} \rangle$

$$X(000, \langle s, s, \overline{s} \rangle) = \mathsf{mn}(U_0(x) \wedge U_0(y) \wedge U_0(z) \wedge s(x, y) \wedge s(y, z) \wedge s(z, x)),$$
where
$$U_0(x) = \exists yzR, \quad U_0(y) = \exists x(x = y \wedge U_0(x)), \quad U_0(z) = \exists x(x = z \wedge U_0(x)).$$

The other cases are similar, we leave checking them as an exercise.[32]

Finally, to show that $\mathsf{mn}(D(x)) = \{\langle a_0, b, c \rangle : b, c \in M\} \notin A$, observe that the domain of each element in B either contains U_0 or else is disjoint from it, and therefore the same holds for their unions. This shows that $\mathsf{mn}(D) \notin A$, i.e., D cannot be explicitly defined in $\mathfrak{M}$. Since $\mathfrak{M}$ is a model of Th, this means that $\Sigma(D)$ is not equivalent to any explicit definition that contains only 3 variables. □

Remark 2.1 The variant of $\mathcal{L}_n$ in which we allow only models of size $\leq n+1$ has the strong BDP, for all n, see [1]. Another variant of $\mathcal{L}_n$ that has the strong BDP is when we allow models of all sizes but in a model truth is defined by using only a set of selected (so-called admissible) evaluations of the variables (a generalized model then is a pair consisting of a model in the usual sense and this set of admissible evaluations). For more on this see [2]. □

We note that $\mathcal{L}_2$ does not have the strong BDP (this is proved in [1]), and we do not know whether it has the weak BDP. $\mathcal{L}_1$ has the strong BDP.

Theorem 2.6 implies that Craig's Interpolation Theorem does not hold for n-variable logic, either, for $n \geq 3$. This is so because in the proof of the weak Beth Definability Theorem we constructed the explicit definition from an interpoland.

3 From pure FOL to FOL with equality

What is equality? How is equality introduced? What are functions, constants? How can they be defined? (What are the rules for defining them?)

[32] Exercise 5.25.

Conventions for talking about partial functions. What is many-sorted logic? What are sorts? How can they be defined? FOL with dependent sorts (FOLDS). These all can be considered as "syntactic sugars", tools for simplifying our formulas. In this section we deal, briefly, with equality only. We address the question: What makes two things equal/identical?

To investigate equality, let's start out from FOL without equality, sometimes called *basic* or *pure FOL*. We have relation symbols $R_1, ...$ of finite arities $n_1, ...$ The logical connectives are "or", "not", and "exists", in symbols $\vee, \neg, \exists$. (We consider the other often used logical connectives $\wedge, \rightarrow, \leftrightarrow, \forall$ as derived, compound ones.) Formulas, models, evaluations of variables and the satisfaction relation are as usual. (An even more basic, so-called *restricted FOL* is where only the atomic formulas $R(v_0, ..., v_{n-1})$ are allowed in place of all the atomic formulas $R(v_{i_1}, ..., v_{i_n})$.)

To understand equality, we now use basic FOL to "talk about equality". We add a new special two-place relation symbol $\equiv$ to the language and we state the following axioms, called the *equality axioms*. We will write $v_i \equiv v_j$ in place of $\equiv (v_i, v_j)$.

$$x \equiv y \wedge y \equiv z \rightarrow x \equiv z, \quad x \equiv y \rightarrow y \equiv x, \quad \forall x \exists y \, x \equiv y,$$

$$R(x_1, ..., x_n) \wedge x_1 \equiv y_1 \wedge ... \wedge x_n \equiv y_n \rightarrow R(y_1, ..., y_n).$$

The first line says that $\equiv$ forms a partition of the universe, and the second line says that no basic relation of the language differentiates the elements of a block. Let $\mathfrak{M}$ be a model of the extended language $\mathcal{L}^+$ and assume that $\equiv$ satisfies the above equality axioms for the language $\mathcal{L}$ (i.e., the last axiom holds for all the basic relation symbols of $\mathcal{L}$). For any $a \in M$ let $\overline{a}$ denote the block of $\equiv$ containing a, and let $\overline{\mathfrak{M}}$ denote the structure $\mathfrak{M}$ factored with $\equiv$, i.e.

$$\overline{a} = \{b \in M : a \equiv b\}, \quad \overline{M} = \{\overline{a} : a \in M\},$$

$$\overline{R} = \{\langle \overline{a_1}, ..., \overline{a_n} \rangle : \langle a_1, ..., a_n \rangle \in R\} \quad \text{and} \quad \overline{\mathfrak{M}} = \langle \overline{M}, \overline{R} : R \in \mathcal{L}^+ \rangle.$$

Theorem 3.1 *With using the above notation, let $\varphi(x_1, ..., x_n) \in \mathcal{L}^+$ be arbitrary, and let $a_1, ..., a_n \in M$. Then*

$$\mathfrak{M} \models \varphi(a_1, ..., a_n) \quad \Leftrightarrow \quad \overline{\mathfrak{M}} \models \varphi(\overline{a_1}, ..., \overline{a_n}).$$

The proof of the above theorem goes by an easy induction. $\square$

In the factored structure $\overline{\mathfrak{M}}$ the relation $\equiv$ denotes the identity relation $\{\langle m, m\rangle : m \in \overline{M}\}$. The passage from $\mathrm{FOL}^{\neq}$ to $\mathrm{FOL}^{=}$ consists of declaring $\equiv$ to belong to the logic as a *logical binary relation symbol* with the fixed meaning in all models as the identity relation. Thus $\mathrm{FOL}^{=}$ is basically $\mathrm{FOL}^{\neq}$ together with a binary relation symbol $\equiv$ for which the axioms of equality are postulated. For convenience, we declare that the interpretation of this $\equiv$ should be the identity relation, this belongs to the logic, so when we specify a model, we do not have to give the meaning of $\equiv$ because it is provided by logic as being the identity relation. Finally, we denote $\equiv$ by $=$.

We give an example. We want to get the notion of sets from the notion of sequences by disregarding the order and number of occurrence among the members of a sequence. For simplicity, assume that we have binary relations R_i in our language (with the intended meaning of $R_i(x, y)$ being "the i-th member of the sequence x is y"). Define $x \equiv y$ to mean that "the ranges of x and y are the same", i.e.,

$$x \equiv y \quad \Leftrightarrow \quad \forall z(\exists i R_i(x, z) \leftrightarrow \exists j R_j(y, z)).$$

Then define the "element of relation" E as follows:

$$xEy \quad \Leftrightarrow \quad \exists x'(x' \equiv x \wedge \exists i R_i(y, x')).$$

Let's forget the relations R_i and keep only the new E in our language. Now, $\equiv$ satisfies the axioms of equality wrt. E. Two "sets" may have the same elements, thus equal, and yet not identical because we "did not erase the information of the original sequence structure". If we want to be thorough in our formation of a new concept of set, we may want to say that "the identity of a set is given by its elements and by nothing else". This means that we factor out by the equivalence relation $\equiv$ and replace it with the "true identity" $=$. The factor structure satisfies the so-called *Axiom of Extensionality*:

$$\forall z(z \in x \leftrightarrow z \in y) \rightarrow x = y.$$

We usually state an axiom of extensionality when we want to emphasize that the concept in question (in our case, set) is determined by the properties in question (in our case, its elements).

4 Dynamics of concept formation

In this section we investigate connections between FOL theories of different languages. What makes two FOL languages different? Their vocabularies. The *vocabulary* (sometimes called also similarity type) of a FOL language consists of the concepts (together with their arities or ranks) we do not analyze further in the given language. We can refine or revise this choice of basic concepts by changing the language via the use of interpretations, see section 4.5.

By a theory in a language $\mathcal{L}$ we understand a set of sentences in $\mathcal{L}$, but we will be interested in the set of its consequences

$$\mathsf{Th}(T) = \{\varphi \in \mathcal{L} : T \models \varphi\}.$$

Two theories on the same languages are said to be *equivalent* iff their consequences are equal

$$T \equiv T' \iff \mathsf{Th}(T) = \mathsf{Th}(T').$$

When we want to indicate the language $\mathcal{L}$ of which we consider T to be a theory of, we write $\mathcal{L}(T)$ for $\mathcal{L}$. In this section, as before, in the definitions we will assume that the languages have only relation symbols, but in the examples we will use function and constant symbols, too.

4.1 Definitional extension

Definitional extension of a language is introducing notation, definitions for ease of talk. Technically, definitional extension of a language (and of a theory) consists of adding some new relation symbols together with explicit definitions to them. Let $\Sigma(\underline{R})$ be a set of explicit definitions for a sequence of relation symbols not in (the vocabulary of) $\mathcal{L}(T)$. Then we say that $T \cup \Sigma(\underline{R})$ is a definitional extension of T. More precisely, we say that T' is a *definitional extension* of T, when T' is equivalent to $T \cup \Sigma$ for some set Σ of explicit definitions for relation symbols not in $\mathcal{L}(T)$.[33] In symbols, we denote this by $T \xrightarrow{\Delta} T'$, or by $T' \xleftarrow{\Delta} T$.

There is a strong connection between a definitional extension of a language and the unextended language: we can consider the new formulas as

[33] We always assume, implicitly, that only one definition is given for a relation symbol in Σ.

being abbreviations of old formulas. A tangible formulation of this is to specify a translation function tr from the extended language to the original one, and then one can think of a new formula φ as being an abbreviation for $\mathsf{tr}(\varphi)$. This means that after having introduced these new symbols, we can eliminate them at any time we want.

We now specify this translation function. The idea is that we replace $R(v_0, ..., v_n)$ with its explicit definition, we replace the same atomic formula but with a different sequence of variables $R(x_0, ..., x_n)$ by the corresponding version of φ_R we get by using Tarski's substitution of variables, and otherwise we leave the logical structures of the formulas as they were.[34]

Definition 4.1 *(Eliminating explicit definitions)*

$\mathsf{tr}(R(v_0, ..., v_n)) = \varphi_R$ *if* $R(v_0, ..., v_n) \leftrightarrow \varphi_R$ *is in* Σ.

$\mathsf{tr}(R(x_0, ..., x_n))$ *is the appropriate substituted version of* φ_R

$\mathsf{tr}(S(x_1, ..., x_n)) = S(x_1, ..., x_n)$ *if* $S(x_1, ..., x_n) \in \mathcal{L}$,

$\mathsf{tr}(v_i = v_j) = v_i = v_j$,

$\mathsf{tr}(\neg\varphi) = \neg\mathsf{tr}(\varphi)$, $\mathsf{tr}(\varphi \vee \psi) = \mathsf{tr}(\varphi) \vee \mathsf{tr}(\psi)$, $\mathsf{tr}(\exists v_i\varphi) = \exists v_i\mathsf{tr}(\varphi)$. □

Theorem 4.1 $\Sigma \models \varphi \leftrightarrow \mathsf{tr}(\varphi)$ *for every formula* φ *in the extended language. Also,* $\mathsf{tr}(\varphi) = \varphi$ *for every formula in the unextended language.* □

Definitional extension preserves many properties of theories, e.g., it preserves finite axiomatizability, decidability, expressiveness (the same concepts can be expressed): it leaves the "content" of the theory unchanged. However,

[34]On the translation of $R(x_0, ..., x_n)$. The most natural way would be to define the translation of $R(x_0, ..., x_n)$ as $\exists v_0(v_0 = x_0 \wedge ... \wedge \exists v_n(v_n = x_n \wedge \varphi_R)$. However, there may be "collisions of variables" that would make the translated formula not to mean what we want. E.g., from the formula $R(y, x)$ we would get by this kind of substitution $\exists x(x = y \wedge \exists y(y = x \wedge R(x, y))$ which is semantically equivalent to $R(y, y)$ and not to $R(y, x)$ as we would like. Therefore, to specify the translation of the substituted new atomic formulas $R(x_0, ..., x_n)$ we use auxiliary sequences $y_0, ..., y_n$ of new variables to avoid this kind of "collision of variables". We can use any pre-agreed such new y's that make the job. E.g., we can agree that $y_0, ..., y_n$ is suitable for $v_{i_0}, ..., v_{i_n}$ if y_i is v_{m+i} where m is the maximum of $i_0, ..., i_n, 0, ..., n$ if x_i is different from v_i, otherwise let y_i be v_i. After this, let's specify "the appropriate substitution of φ" as $\exists v_0(v_0 = y_0 \wedge ... \wedge \exists v_n(v_n = y_n \wedge \exists y_0(y_0 = x_0 \wedge ... \wedge \exists y_n(y_n = x_n \wedge \varphi))))$.

definitional extension introduces convenience in expressing, we can express the same thing shorter, clearer, maybe in nicer form: it enriches the ways we can say things.

Example 4.1 *The language of Zermelo-Fraenkel set theory* ZF *contains one binary relation symbol, the "elementhood" relation* ϵ*. When working in* ZF*, we use many explicitly defined concepts (one has the feeling that we use them more extensively than* ϵ *itself). For example, we use* $x \subseteq y$ *as an abbreviation for* $\forall z(z \in x \to z \in y)$*, we use* $x \cap y = z$ *as an abbreviation for* $\forall v(v \in z \leftrightarrow [v \in x \wedge v \in y])$*, we use* $\emptyset$ *for the unique set which has no elements at all, we use* $\{x, y\}$ *for the set which has* x, y *as elements and has no other elements, etc. Should we translate and use all these formulas to the original language containing only* ϵ*, we would go crazy and would abandon the nice simple language of* ZF*.*

Example 4.2 *If we add a relation symbol to a language* without an explicit definition*, we increase the expressive power. For example, let's add a new unary relation symbol* R *to the language of the reals. This increases the expressive power: the FOL-theory of the reals is decidable, while the FOL-theory of the reals on the so extended language becomes undecidable. The reason for this is that on the extended language we can talk about subsets of the reals, while on the original language we cannot do so. For example, we can say that* $[R(0) \wedge \forall x(R(x) \to R(x+1)] \to \varphi$*. When* φ *is existential and talks about the elements of* R *only (i.e., when all the quantifiers are restricted to elements of* R*), this formula will be valid in the extended theory of the reals exactly when* φ *is valid for the integers. Solvability of Diophantine equations in the integers is undecidable, this implies that the theory of the reals on the extended language is undecidable. This example also shows that the integers cannot be defined within the reals (on the original, unextended language).*

Example 4.3 *Groups sometimes are defined as structures with one associative, cancellative operation* $+$ *which has a zero-element* $(x + y = x + z \to y = z, \quad y + x = z + x \to y = z, \quad \forall x \exists y\, x + y + z = y + x + z = z)$*. Groups at other times are defined as structures with an associative binary operation* $+$*, a unary operation* $-$ *and a constant* 0 *satisfying the equations* $x + 0 = 0 + x = x, \quad x + -x = -x + x = 0$*. The secondly defined class is a definitional extension of the first class, as explicit definitions we can take* $0 = z \leftrightarrow \forall x\, x + z = x, \quad -x = z \leftrightarrow 0 = z + x$*. An advantage of the second*

definition is that it consists of equations only, and therefore we can use the powerful methods of universal algebra. E.g., subalgebras, homomorphic images, direct products of groups with operations $+, -, 0$ are again groups, while a subalgebra of a group defined in the first way may not be a group (since we may omit the zero-element).

4.2 Definitional equivalence

Definitionally equivalent theories have the same content, but they have different "cloths". We get a definitionally equivalent theory from one by introducing new relations (via explicit definitions) and at the same time forgetting (leaving out) ones that can be expressed by the use of these newly introduced relations. Definitionally equivalent theories may look rather different, it may come as surprise that they are in fact equivalent, see, e.g., section 4.3. Technically, definitional equivalence is the symmetric and transitive closure of the notion of definitional extension. Thus definitional equivalence of theories preserves all the properties a definitional extension does.

Definition 4.2 *Two theories T, T' are said to be* definitionally equivalent*, in symbols $T \stackrel{\Delta}{\equiv} T'$, if there is a sequence $T_1, ..., T_n$ of theories such that $T = T_1$, $T' = T_n$, and for all $1 \leq i < n$ either $T_i \stackrel{\Delta}{\longrightarrow} T_{i+1}$ or $T_i \stackrel{\Delta}{\longleftarrow} T_{i+1}$.* □

Example 4.4 *Renaming the relation symbols occurring in T to completely new symbols results a theory T' definitionally equivalent to T. Indeed, let T'' be the union of T and T', then T'' is a definitional extension both of T and of T'.*

The next theorem says that when the vocabularies of definitionally equivalent theories T, T' are disjoint, the chain leading from T to T' required for showing definitional equivalence can always be taken to have three elements only. Thus, when T, T' are arbitrary and definitionally equivalent, there is always a 5-long chain of definitional extensions between them (by Example 4.4). We call a function f from one language to another *structural* if f is the translation function associated to an explicit definition as in Def.4.1. Let $\mathsf{Mod}(T)$ denote the class of all models of T.

Theorem 4.2 *(*characterizations of definitional equivalence*) Assume that T and T' have disjoint vocabularies. Then the following (i)-(iv) are equivalent.*

(i) $T \stackrel{\Delta}{\equiv} T'$

(ii) *T and T' have a common definitional extension, i.e., there is a theory T'' such that*

$$T \quad \stackrel{\Delta}{\longrightarrow} \quad T'' \quad \stackrel{\Delta}{\longleftarrow} \quad T'.$$

(iii) *There are structural translation functions $\mathsf{tr} : \mathcal{L}(T) \to \mathcal{L}(T')$ and $\mathsf{tr}' : \mathcal{L}(T') \to \mathcal{L}(T)$ which are inverses of each other w.r.t. T and T', i.e., for all $\varphi \in \mathcal{L}$ and $\psi \in \mathcal{L}'$ we have*

$$T \models \varphi \leftrightarrow \mathsf{tr}'\mathsf{tr}\varphi \qquad \text{and} \qquad T' \models \psi \leftrightarrow \mathsf{trtr}'\psi.$$

(iv) *There is a bijection β between $\mathsf{Mod}(T)$ and $\mathsf{Mod}(T')$ that is defined along two explicit definitions Σ and Σ' the following way: if $\mathfrak{M} \models T$ and $\mathfrak{M}' = \beta(\mathfrak{M})$ then the universes of $\mathfrak{M}$ and $\mathfrak{M}'$ are the same, the relations in $\mathfrak{M}'$ are the ones defined in $\mathfrak{M}$ according to Σ and vice versa, the relations in $\mathfrak{M}$ are the ones defined in $\mathfrak{M}'$ according to Σ'.*

Various textbooks define the notion of definitional equivalence of theories in various ways. Definitional equivalence of theories with disjoint vocabularies is defined as in Thm.4.2(ii) in, e.g., [20, p.42], [14, pp.60,61], [17, section 6], definitional equivalence is defined as in Thm.4.2(iv) e.g., in [13, p.56](iv) and generalizations of this latter definition can be found in [5]. These definitions are equivalent by the above Thm.4.2.

Definitional equivalence gives a "new cloth", new appearance for a theory. Definitionally equivalent theories are two presentations of the same theory. They differ only in the choices in their basic vocabularies.

Example 4.5 *Boolean algebras as ordering (having one binary relation $\leq$) can be defined as strongly complemented bounded lattices. This means that $\leq$ is an ordering in which every two elements have a supremum (least upper bound) and an infimum (greatest lower bound), in every interval each element has a unique complement, and there are a lowest element and a greatest one. This definition is very convenient when we want to visualize Boolean algebras. Another way of defining Boolean algebras is by defining them as structures having 2 binary functions $+, \cdot$, one unary function $-$ and two constants $0, 1$*

and postulating some equations for these. These two versions have disjoint vocabularies but they are definitionally equivalent.

We sketch a proof of the above definitional equivalence. We use Thm.4.2(iii). Let $\mathsf{Th}(\leq)$ *and* $\mathsf{Th}(+,\cdot,0,1,-)$ *denote the sets of the above defining formulas. Let us choose the two sets of explicit definitions as*

$$\Sigma(\leq) := \{x \leq y \leftrightarrow x \cap y = y\},$$

$$\begin{aligned}\Sigma(+,\cdot,0,1,-) := \\ &\{x + y = z \leftrightarrow [z \geq x \wedge z \geq y \wedge \forall v(v \geq x \wedge v \geq y \rightarrow v \geq z)],\\ &x \cdot y = z \leftrightarrow [z \geq x \wedge z \leq y \wedge \forall v(v \leq x \wedge v \leq y \rightarrow v \leq z)],\\ &0 = z \leftrightarrow \forall v(z \leq v), \quad 1 = z \leftrightarrow \forall v(z \geq v),\\ &-x = z \leftrightarrow \forall v([v \leq x \wedge v \leq z \rightarrow \forall w\, v \leq w] \wedge [v \geq x \wedge v \geq z \rightarrow \\ &\qquad \forall w\, v \geq w])\}.\end{aligned}$$

Let $\mathsf{tr}_{\leq}$ *and* tr_{+} *be the structural translation functions belonging to these two (sets of) explicit definitions. Now, one can prove the following:*

$$\mathsf{Th}(\leq) \models x \leq y \leftrightarrow \mathsf{tr}_{\leq}\mathsf{tr}_{+}(x \leq y),$$

$$\begin{aligned}&\mathsf{Th}(+,\cdot,0,1,-) \models \varphi \leftrightarrow \mathsf{tr}_{+}\mathsf{tr}_{\leq}\varphi,\\ &\quad \textit{for all } \varphi \in \{x + y = z,\ x \cdot y = z,\ 0 = z,\ 1 = z,\ -x = 2\}.\end{aligned}$$

This verifies definitional equivalence of the two "versions" of Boolean algebras. □

In the next section we describe an example for definitionally equivalent theories in more detail.

4.3 Peano Arithmetic and Finite Set Theory

Let ZF_0 denote Zermelo-Fraenkel Set Theory but with the Axiom of Infinity changed to its negation (and adding explicitly the existence of transitive supsets, since the proof of their existence in ZF needs the axiom of infinity). The sole basic symbol of ZF_0 is the elementhood relation ϵ. The intended (or standard) model of ZF_0 is the collection of hereditarily finite sets with the membership relation $\mathbb{H} = \langle \mathsf{HF}, \in \rangle$. Hereditarily finite sets are sets built explicitly from the empty set such as $\{\{\emptyset\}, \emptyset, \{\{\emptyset\}, \emptyset\}\}$, i.e., sets written up from $\emptyset$ by using the formation of finite sets $\{a_1, a_2, ..., a_n\}$.[35] All the

[35] We can write $\emptyset$ as $\{\}$, then hereditarily finite sets can be identified with finite sequences of equal numbers of { and } beginning with a { and ending with a }.

hereditarily finite sets are definable constants in ZF_0 (as well as in ZF), i.e., they have "names". E.g., the "name" (or, definition as a constant) of $\emptyset$ is "the x for which $\forall y\, y \notin x$", the "name" of $\{\emptyset\}$ is "the x for which $\forall y(y \in x \leftrightarrow y = \emptyset)$", etc. Below we briefly recall the axioms of ZF_0.

Definition 4.3 *(Theory ZF_0 of hereditariy finite sets) The axioms of ZF_0 are the following.*

Extensionality $\forall v(v \in x \leftrightarrow v \in y) \to x = y$.

Existence axioms:

Empty set $\exists z \forall v\, v \notin z$,
Unordered pairs $\exists z \forall v(v \in z \leftrightarrow [v \in x \vee v \in y])$,
Powerset $\exists z \forall v(v \in z \leftrightarrow v \subseteq x)$,
Union $\exists z \forall v(v \in z \leftrightarrow \exists y[y \in x \wedge v \in y])$,
Transitive supsets $\exists z(x \subseteq z \wedge \forall v w[v \in w \wedge w \in z \to v \in z])$,
Separation scheme $\exists z \forall v(v \in z \leftrightarrow [v \in x \wedge \varphi(v)])$.

Regularity (or Foundation) $\exists y(y \in x \wedge y \cap x = \emptyset)$.

Only finite sets $[f : x \to x \wedge$ *f is one-to-one* $] \to$ *f is onto.* □

It turns out that PA and ZF_0 are definitionally equivalent theories, in spite that they look rather different and their intended structures look rather different. (E.g., the usual way we draw these models, $\mathbb{N}$ looks rather narrow, it has a natural linear order, while the set of hereditarily finite sets is rather wide, one does not see at once a natural linear order for $\mathbb{H}$.)

Theorem 4.3 PA *and* ZF_0 *are definitionally equivalent theories.*

Sketch of proof. In this proof we want to highlight the key ideas, for more detail we refer to [20, sec.7.5, sec.7.6] and to [16].

The vocabularies of PA and ZF_0 are $+, \star$ and ϵ, respectively. We will exhibit an explicit definition $\Sigma(\epsilon)$ of ϵ in terms of the basic symbols of PA, and we will provide a set $\Sigma(+, \star)$ of explicit definitions for the basic symbols of PA in terms of ϵ in such a way that the translation functions belonging to these ($\Sigma(\epsilon)$ and $\Sigma(+, \star)$) are inverses of each other.

First we define $\Sigma(\epsilon)$. We design $\Sigma(\epsilon)$ so that it works in the standard model $\mathbb{N} = \langle \omega, +, \star \rangle$ of PA, and then we check that the same definition works in all models of PA. Let $k, n \in \omega$, we have to define when $k\, \epsilon\, n$ should hold.

The idea is that we write up n in the binary system, then we get a finite sequence of 0's and 1's, and we consider this sequence to be the characteristic function of n as the "set of its ϵ-elements". I.e., we define $k\,\epsilon\,n$ iff in the k'th place of the binary form of n there is a 1 (in other words, iff the k'th digit in n's binary form is 1). Example: Let $n = 11$ (i.e., n is eleven), then the binary form of n is 1101 ($2^3 + 2^1 + 2^0$), hence $0\,\epsilon\,11$, $1\,\epsilon\,11$, $2\ \not\epsilon\,11$, $3\,\epsilon\,11$, $4\ \not\epsilon\,11$, in general $k\ \not\epsilon\,11$ for all $k \geq 4$. Thus, according to the definition of ϵ we have $11 = \{0, 1, 3\}$ (in the sense of sets defined from ϵ) and similarly $0 = \emptyset$, $1 = \{\emptyset\}$, $3 = \{\emptyset, \{\emptyset\}\}$, finally

$$7 = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}.$$

Here is an explicit definition $\Sigma(\epsilon)$ for ϵ:

$$k\,\epsilon\,n \leftrightarrow \exists m m' r(n = m \star 2^k + r \wedge r < 2^k \wedge m = 2 \star m' + 1).$$

We used exponentiation 2^k in the above definition, we can do that because we have seen that exponentiation can be explicitly defined in PA. The next claim says that if in any model of PA we add the relation ϵ as defined in $\Sigma(\epsilon)$ above and then forget (i.e., omit) the original operations $+, \star$, we get a model of ZF_0. Let $\mathsf{tr}(\Sigma(\epsilon))$ denote the natural translation function belonging to $\Sigma(\epsilon)$. We state the claim without proof.

Claim 4.1 $\mathsf{PA} \models \mathsf{tr}(\Sigma(\epsilon))(\varphi)$ *for all* $\varphi \in \mathsf{ZF}_0$.

Now we turn to re-defining $+, \star$ from the just defined ϵ. Our plan is to begin with re-defining the ordering $\leq$ between numbers (from ϵ defined as in $\Sigma(\epsilon)$). From $\leq$ we will be able to define suc, and then $+, \star$.

As before, we design the definition of $\leq$ from ϵ so that it works in $\mathbb{N}$, and then we check that the so obtained definition works in all models of PA. How can we see from the binary forms of k and n whether $k \leq n$ or not? If k is smaller-or-equal than n at each binary digit, then surely $k \leq n$, and the former can be formalized in terms of ϵ as $k \subseteq n$. However, this is not the only way k can be smaller than n. Assume that $k \leq n$ but $k \not\subseteq n$. Then there is u such that $u\,\epsilon\,k \wedge \neg u\,\epsilon\,n$. This means that the u'th digit is 1 in k's binary form while it is 0 in n's binary form. Then $k \leq n$ can still hold if at a later place, say at $v > u$ in the v'th place there is a 0 in the binary form of k while there is a 1 in the binary form of n. So, the following is true in $\mathbb{N}$, and it fact can be proved from PA:

$$x \leq y \quad \leftrightarrow \quad \forall u([u\,\epsilon\,x \wedge \neg u\,\epsilon\,y] \rightarrow \exists v[u \leq v \wedge \neg v\,\epsilon\,x \wedge v\,\epsilon\,y])$$

In particular, we get back that $0 \leq y$ for all y since 0 has no elements at all. The above is not an explicit definition yet, because $u \leq v$ occurs in the right-hand side of the $\leftrightarrow$. However, u and v are lower in the ϵ-hierarchy than x and y (since they are ϵ-elements of x, y respectively). Thus to check whether $u \leq v$ we have to check whether $\leq$ holds between some elements of u, v, etc. This procedure has to end in a finite steps because in finite steps we get to the empty set (by the Axiom of regularity), and we know that 0 is the least element. So, this is kind of a recursive argument. Just as in PA, recursive definitions of the above kind can be made explicit in ZF_0, as follows. We can express pairs by using ϵ as[36]

$$\langle x, y \rangle = \{\{x\}, \{x, y\}\}$$

then we denote

$$w \times w = \{\langle u, v \rangle : u, v \, \epsilon \, w\}$$

finally let $\mathsf{trans}(w)$ denote that ϵ is transitive in w, i.e.,

$$\mathsf{trans}(w) \leftrightarrow \forall uv([u \, \epsilon \, w \wedge v \, \epsilon \, u] \rightarrow v \, \epsilon \, w)$$

Now, the above definition of $\leq$ can be made explicit as

$$\begin{array}{l} x \leq y \quad \leftrightarrow \quad \exists wk(x \, \epsilon \, w \wedge y \, \epsilon \, w \wedge \mathsf{trans}(w) \wedge k \subseteq w \times w \wedge \langle x, y \rangle \, \epsilon \, k \wedge \\ \forall p, q \, \epsilon \, w(\langle p, q \rangle \in k \leftrightarrow \forall u([u \, \epsilon \, p \wedge \neg u \, \epsilon \, q] \rightarrow \exists v[\langle u, v \rangle \, \epsilon \, k \wedge \neg v \, \epsilon \, p \wedge v \, \epsilon \, q])) \end{array}$$

In the above, the relation k plays the same role as a finite sequence imitating "computation" in the method of making recursive definitions explicit in PA.

By using $\leq$ we can define suc as

$$\mathsf{suc}(x) = y \quad \leftrightarrow \quad \forall z(x \leq z \leq y \rightarrow (x = z \vee z = y))$$

then we can define $+, \star$ from suc recursively and make the recursive definition explicit by using the previous technique using a relation k in a transitive set w. Let $\Sigma(+, \star)$ denote the set of explicit definitions for $+, \star$ from ϵ we have just outlined. We state the following claim without proof.

Claim 4.2 $\mathsf{PA} \cup \Sigma(\epsilon) \models \Sigma(+, \star)$.

[36] $\{x\}$ denotes the unique number n for which ($k \, \epsilon \, n$ holds iff $k = x$), etc.

By this we finish the sketch of proof for Theorem 4.3. □

The above definitional equivalence enriches our views both for numbers and finite sets. We can "see" finite sets when seeing numbers, and we can "see" numbers when seeing finite sets. Keeping this connection in mind, some problems may be handled more naturally as dealing with numbers, and some as dealing with finite sets.

4.4 Concept algebra of a theory

We have talked about contents versus appearances of theories. What is the "content" of a theory? The concept algebra $CA(T)$ of a theory T represents the "content" of a theory T, stripped of syntax. Two theories on perhaps different languages are definitionally equivalent iff their concept algebras are isomorphic. An interpretation of T into T' corresponds to a homomorphism from $CA(T)$ into $CA(T')$, and vice versa, each such homomorphism arises from an interpretation of T into T'.

4.5 Interpretations between theories

Importance of breaking up a big theory into many smaller ones. Harvey Friedman's paper about the nature of foundational thinking. Vienna Circle dream.

What are the basic symbols in a FOL language? Explicit definitions are the tools for changing basic notions (into compound ones). Interpretations are connections between languages that have different basic symbols, the connection is established via explicit definitions. Replacing a big theory with a hierarchy of small theories and interpretations between them. This is absolutely necessary when applying logic in other branches of science such as physics, sociology, computer science etc.

Omitting/relaxing axioms from a theory increases the number of concepts: the weaker a theory is the more concepts it has. Weakening a theory is part of understanding it. Example: reverse mathematics. Weak theories are important, it is not a goal to always use strong theories.

We can view an interpretation in two ways. 1) We define theoretical notions, and if they prove to be useful, we "elevate them" to the rank of basic symbols. This way we can make the theory more elegant, more perspicuous. 2) When we investigate a phenomenon, we decide the level of basic notions.

Later, we may want to investigate the notions we decided to be basic in more detail. An interpretation makes a basic symbol into a defined one.

Interpretations can be used to talk about the wrapping-content duality. I.e., what property in a theory comes from the tools we use for expressing it and what do belong to its content. Example.

5 Exercises

Exercise 5.1 *Define a translation function from the language $\mathcal{L}$ of* ZF *set theory containing only one binary relation symbol $\in$ to the one that is enriched with a new constant symbol $\emptyset$. By such a translation we mean that each formula φ of the extended $\mathcal{L}$ should be equivalent to its translation $\mathsf{tr}(\varphi) \in \mathcal{L}$ modulo* ZF *plus the definition of $\emptyset$, i.e., $\mathsf{ZF} + \forall x(x = \emptyset \leftrightarrow \neg\exists y(y \in x)) \models \varphi \leftrightarrow \mathsf{tr}(\varphi)$.*

Exercise 5.2 *Is "the number which is the sum of all the numbers smaller than it" a definition (modulo Peano's Arithmetic* PA*)?*

Exercise 5.3 *Give an explicit definition for $\{n \in \omega : n < 1000\}$ in the language containing $0, \mathsf{suc}$ that works in the theory*[37] *of $\langle \{n \in \omega : n \leq 10000\}, 0, \mathsf{suc}\rangle$ (where* suc *is modified so that $\mathsf{suc}(10000) = 10000$). Give implicit definitions, too.*

Exercise 5.4 *Give an explicit definition for the number-theoretic function* factorial *that works in Peano's Arithmetic! By Theorem 1.1, there is one. (Hint: use the fact that in* PA *being a finite sequence is expressible. So you can use the notion of finite sequences in your definition.)*

Exercise 5.5 *Show that an explicit definition is always an implicit definition, modulo any theory* Th*.*

Exercise 5.6 *Prove Theorem 1.2. Hint: Repeat the proof of Theorem 1.1 with appropriate modifications.*

Exercise 5.7 *Use Henkin's method outlined in the proof of Theorem 1.1 to prove the (strong) completeness theorem of FOL. Namely, prove that if $\Sigma \models \varphi$ then there is a proof that derives φ from Σ. For this purpose, you can use any proof system that you find in a book, or you can even devise your own favorite proof system for this goal.*

[37] By the theory $\mathsf{Th}(\mathfrak{M})$ of a model $\mathfrak{M}$ we understand the set of all sentences true in $\mathfrak{M}$.

Exercise 5.8 *Show that* $\Sigma(\mathsf{R})$ *is a weak but not strong definition of* R *in* Th *whenever* $\mathsf{Th} \cup \Sigma(\mathsf{R})$ *is inconsistent*[38] *and* Th *is consistent.*

Exercise 5.9 *Show that the above kinds of definitions are the only weak but not strong definitions when* Th *is a complete theory.*[39] *I.e., assume that* Th *is complete,* $\Sigma(\mathsf{R})$ *is a weak definition w.r.t.* Th*, and* Th *has at least one model in which there is an* R *satisfying* $\Sigma(\mathsf{R})$*. Then* $\Sigma(\mathsf{R})$ *is also a strong definition w.r.t.* Th*. Hint: Use Theorem 1.2.*

Exercise 5.10 *Show that if* $\Sigma(\mathsf{R})$ *is a weak definition in* Th*, then the class of models of* Th *in which* $\Sigma(\mathsf{R})$ *has a "solution for* R*" is axiomatizable.*

Exercise 5.11 *Assume that* $\mathfrak{M}$ *and* $\mathfrak{N}$ *are elementarily equivalent*[40] *models of* $\mathcal{L}$ *and* $\Sigma(\mathsf{R})$ *has a solution in* $\mathfrak{M}$ *and it does not have a solution in* $\mathfrak{N}$*. Then* $\Sigma(\mathsf{R})$ *is no weak definition for any theory* Th *which* $\mathfrak{M}$ *is a model of.*

Exercise 5.12 *Assume that* $\Sigma(\mathsf{R}, \mathsf{B})$ *is a weak implicit definition for the pair of* R, B*, i.e.,* $\Sigma(\mathsf{R}, \mathsf{B})$ *holds only for one pair or relations in each model. Show that then* R, B *have explicit definitions, i.e., there are formulas* $\varphi_{\mathsf{R}}, \varphi_{\mathsf{B}}$ *containing neither* R *nor* B *such that* $\Sigma(\mathsf{R}, \mathsf{B}) \models (\mathsf{R} \leftrightarrow \varphi_{\mathsf{R}}) \wedge (\mathsf{B} \leftrightarrow \varphi_{\mathsf{B}})$*. (Hint: Use Theorem 1.2 twice.)*

Exercise 5.13 *Show that the ordering* $\leq$ *is not definable from successor in* $\langle \omega, \mathsf{suc} \rangle$*.*

Exercise 5.14 *Define* suc *from addition explicitly. I.e., write up a formula* $\varphi(x, y)$ *in the FOL language of* $+$ *such that* $\langle \omega, \mathsf{suc}, + \rangle \models \forall xy(\mathsf{suc}(x) = y \leftrightarrow \varphi(x, y))$*.*

Exercise 5.15 *Define* suc *from ordering explicitly. I.e., write up a formula* $\varphi(x, y)$ *in the language of* $\leq$ *such that* $\langle \omega, \mathsf{suc}, \leq \rangle \models \forall xy(\mathsf{suc}(x) = y \leftrightarrow \varphi(x, y))$*.*

---

[38] A theory is consistent iff it has at least one model.

[39] A theory is complete in $\mathcal{L}$ if it implies either φ or $\neg\varphi$, but not both, for all sentences φ in $\mathcal{L}$.

[40] Two models are said to be elementarily equivalent if they are not distinguishable by a formula, i.e., if their theories are the same.

Exercise 5.16 *Show that addition cannot be defined from multiplication in $\langle \omega, \star \rangle$, i.e., show that for no formula $\varphi(x,y,z)$ in the language of $\star$ is it true that $\mathbb{N} \models x + y = z \leftrightarrow \varphi(x,y,z)$. Hint: Use that each permutation of the prime numbers induces an automorphism of $\langle \omega, \star \rangle$.*

Exercise 5.17 *Show that multiplication as well as addition can be defined from exponentiation. I.e., give two formulas φ, ψ in the language of $\langle \omega, \exp \rangle$ for which*

$$\langle \omega, +, \star, \exp \rangle \models \forall xyz[(x \star y = z \leftrightarrow \varphi(x,y,z)) \wedge (x + y = z \leftrightarrow \psi(x,y,z))] \ .$$

Exercise 5.18 *Prove $\forall xyx'y'(\mathsf{pair}(x,y) = \mathsf{pair}(x',y') \rightarrow (x = x' \wedge y = y')$ from* PA*.*

Exercise 5.19 *Show that 3 is the smallest number which is not a pair. What are the next two smallest numbers which are not pairs?*

Exercise 5.20 *Prove* $\mathsf{PA} \models \forall xy \exists s(\mathsf{mem}(s,0,x) \wedge \mathsf{mem}(s,1,y))$.

Exercise 5.21 *Prove that $\Delta(\exp)$ is an implicit definition in $\mathbb{N}$. Prove that $\Delta(\exp)$ is an implicit definition in* PA *also.*

Exercise 5.22 *Can in the definition of* $\mathsf{upvar}(x,n)$ *the quantifier $\exists si$ be changed to $\forall si$?*

Exercise 5.23 *As a "programming exercise" define, similarly to* $\mathsf{upvar}(x,n)$*, a formula* $\mathsf{maxvar}(x,n)$ *with the meaning that v_n is the variable with maximal index occurring in the formula φ for which $x = \ulcorner \varphi \urcorner$. Can you define a similar formula* freevar *for listing all the free variables of a formula?*

Exercise 5.24 *Define a notion of pairs such that each number be a pair! Hint: Enumerate recursively $\omega \times \omega$ and then give an implicit definition for this enumeration via using the notion of sequences defined in the lectures.*

Exercise 5.25 *Prove that $X(ijk,e) \in A$ when ijk is not repetition-free and e is a good choice for ijk. Cf. p.33 in the proof of Theorem 2.6.*

Exercise 5.26 *Give an explicit definition for $D(x)$ that is defined implicitly in the proof of Theorem 2.6. Can you do it by using only four variables?*

Exercise 5.27 *Give two formulas φ, ψ each containing at most 3 variables such that $\models \varphi \rightarrow \psi$ but any interpoland for them has to use more than 3 variables (i.e., if $\models \varphi \rightarrow \chi$ and $\models \chi \rightarrow \psi$, then if χ uses only basic symbols occurring both in φ and ψ then χ uses more than 3 variables).*

6 Solutions for the Exercises

Solution for Exercise 5.1: Define $\mathsf{tr}(\varphi) = \forall x(\neg\exists y\, y \in x \to \varphi(\emptyset/x))$, where x does not occur in φ.

Solution for Exercise 5.2: Yes, the number 3 is the only number which is the sum of all the numbers smaller than it. Indeed, 0 is not the sum of all the numbers smaller than 0 because there is no number smaller than 0. (We understood the sum of the empty set to be undefined.) $1 \neq 0, 2 \neq 1, 3 = 3$, and from here on the number n is always smaller than the sum of the numbers smaller than it. This can be proven from PA. For this provability, we have to express, in the language of PA, the property of n that it is the sum of all the numbers smaller than it. First we define, by recursion, the number $s(n)$ which is the sum of all the numbers smaller than n, and then add $n = s(n)$, i.e., let $\Sigma = \{s(1) = 0, s(n+1) = s(n) + n, \mathsf{R}(n) \leftrightarrow n = s(n)\}$. Now, this Σ defines two new symbols simultaneously, the unary function s and the unary relation symbol R. We can "eliminate" s from Σ by replacing its recursive definition by an explicit one (relying on the fact that finite sequences can be expressed in PA, see the solution for Exercise 5.4).

Solution for Exercise 5.3: For an explicit definition you can take $R(n) \leftrightarrow (n = 0 \vee n = \mathsf{suc}(0) \vee ... \vee n = \mathsf{suc}^{999}(0))$. The following is an implicit definition for the same: $\mathsf{suc}^{999}0 \in \mathsf{R}$, $\mathsf{suc}(n) \in \mathsf{R} \to n \in \mathsf{R}$, $\mathsf{suc}^{1000}0 \notin \mathsf{R}$, $n \notin \mathsf{R} \to \mathsf{suc}(n) \notin \mathsf{R}\}$. We note that this implicit definition does not work in $\langle \omega, 0, \mathsf{suc}\rangle$, because $\mathsf{Th}\langle \omega, \mathsf{suc}\rangle$ has a model in which the above implicit definition has two different solutions.

Solution for Exercise 5.4: $x = \mathsf{factorial}(n) \leftrightarrow \exists s\ [\mathsf{finite\text{-}sequence}(s) \wedge \mathsf{length}(s) \geq n \wedge s_0 = 1 \wedge \forall y < n\, s_{y+1} = s_y \cdot (n+1) \wedge s_n = x]$.

Solution for Exercise 5.5: Assume that $\Sigma(\mathsf{R})$ is an explicit definition. Then Σ is of form $\{\forall \bar{x}(\mathsf{R}(\bar{x}) \leftrightarrow \varphi)\}$ for some formula φ in which R does not occur. Let $\mathfrak{M}$ be any model. We will show that in $\mathfrak{M}$ there is exactly one relation R that satisfies Σ in $\mathfrak{M}$. Indeed, if R satisfies Σ in $\mathfrak{M}$, then R has to be $\{\langle \bar{a} \in M^n : \mathfrak{M} \models \varphi(\bar{a})\}$; and this relation satisfies Σ in $\mathfrak{M}$. Thus Σ has exactly one solution in each model of Th, i.e., it is a(n implicit) definition in Th.

Solution for Exercise 5.6: You can find this proof in [8].

Solution for Exercise 5.7: You can rely on [8, section 2.1].

Solution for Exercise 5.8: Assume that $\mathfrak{M}$ is any model of Th. There is no solution of $\Sigma(\mathsf{R})$ in $\mathfrak{M}$ by our assumption that $\mathsf{Th} \cup \Sigma(\mathsf{R})$ is inconsistent. Thus Σ is a weak definition in Th. There is a model $\mathfrak{M}$ of Th by our assumption that Th is consistent. Σ has no solution in this model $\mathfrak{M}$, by the above. Hence Σ is not a (strong) definition of R in Th.

Solution for Exercise 5.9: By Theorem 1.2, and by our assumptions, there is an explicit definition φ for R in Th, i.e., $\mathsf{Th} \cup \Sigma(\mathsf{R}) \models \mathsf{R} \leftrightarrow \varphi$. Let $\sigma(\mathsf{R})$ be any formula in Σ. Then $\mathsf{Th} \not\models \neg\sigma(\mathsf{R}/\varphi)$, since Th has a model in which Σ has a solution. Then $\mathsf{Th} \models \sigma(\mathsf{R}/\varphi)$ because Th is assumed to be complete and $\sigma(\mathsf{R}/\varphi)$ is a formula in its language. Then $\mathsf{Th} \models \Sigma(\mathsf{R}/\varphi)$, i.e., Σ is a strong definition of R.

Solution for Exercise 5.10: Let φ be the explicit definition of R which exists by the Beth theorem. I.e., $\mathsf{Th} \cup \Sigma(\mathsf{R}) \models \mathsf{R} \leftrightarrow \varphi$. Then in each model of $\mathsf{Th} \cup \Sigma(\mathsf{R}/\varphi)$ there is a unique solution for $\Sigma(\mathsf{R})$. On the other hand, if in a model of Th some element $\sigma(\mathsf{R}/\varphi)$ of $\Sigma(\mathsf{R}/\varphi)$ does not hold, then in this model there can be no solution for $\Sigma(\mathsf{R})$. Thus, the class of models of Th in which $\Sigma(\mathsf{R})$ has a solution is axiomatized by the set $\Sigma(\mathsf{R}/\varphi)$.

Solution for Exercise 5.11: By Exercise 5.10, if Σ is a weak definition of R in Th, then the class of models of Th in which Σ has a solution is axiomatizable. Hence, either both of $\mathfrak{M}$ and $\mathfrak{N}$ are in this class, or neither of them are in this class if they are elementarily equivalent.

Solution for Exercise 5.12: Consider the language $\mathcal{L}$ enriched with the relation symbol R. Then $\Sigma(\mathsf{R}, \mathsf{B})$ is a weak definition of B on this extended language. Hence there is an explicit definition $\psi(\mathsf{R})$ on the extended language for B, i.e., $\Sigma(\mathsf{R}, \mathsf{B}) \models \mathsf{B} \leftrightarrow \psi(\mathsf{R})$. Now, $\Sigma(\mathsf{R}, \psi(\mathsf{R}))$ is a weak definition for R. Therefore, there is φ_{R} such that $\Sigma(\mathsf{R}, \psi(\mathsf{R})) \models \mathsf{R} \leftrightarrow \varphi_{\mathsf{R}}$. Then φ_{R} and $\psi(\mathsf{R}/\varphi_{\mathsf{R}})$ are explicit definitions for R and B implied by $\Sigma(\mathsf{R}, \mathsf{B})$.

Solution for Exercise 5.13: The proof is similar to the proof of Thm.2.1. Add two new constants c, d to the language of successor and let Th be the following theory:

$$\mathsf{Th}(\langle \omega, \mathsf{suc} \rangle) \cup \{c \neq \mathsf{suc}^n(0) \wedge c \neq \mathsf{suc}^n(d) \wedge d \neq \mathsf{suc}^n(0) \wedge d \neq \mathsf{suc}^n(c) : n \in \omega\}.$$

It is easy to show that every finite subset of Th is consistent: let T_0 be any finite subset of Th, let k be the maximal number for which suc^k occurs in Th_0. Then if we take c, d far apart from each other and from 0, e.g., we take them to be $k+1, 2k+2$ respectively, the structure $\langle \omega, \mathsf{suc}, c, d\rangle$ satisfies Th_0. Let $\mathfrak{M}$ be any model of Th. Since the theory of successor is contained in Th, this $\mathfrak{M}$ consists of one island like ω and some islands like the integers $\mathbb{Z}$. Now, Th says that c, d lie in two different $\mathbb{Z}$-islands. Let $f : M \to M$ be a permutation of M which interchanges c and d and respects suc. Assume now that $\leq$ can be defined by a formula $\varphi(x, y)$ in the language of suc, i.e.,

$$\langle \omega, \mathsf{suc}\rangle \models \forall xy(x \leq y \leftrightarrow \varphi).$$

Then $\mathsf{Th}(\langle \omega, \mathsf{suc}\rangle) \models \forall xy[(\varphi(x, y) \vee \varphi(y, x) \wedge (\varphi(x, y) \wedge \varphi(y, x) \to x = y))]$. So, assume that $\mathfrak{M} \models \varphi(c, d)$. Then $\mathfrak{M} \models \varphi(d, c)$ because of the permutation f that interchanges c and d and respects suc. But then $c = d$ should be the case by $\mathsf{Th}(\langle \omega, \mathsf{suc}\rangle) \models \forall xy[(\varphi(x, y) \wedge \varphi(y, x) \to x = y))]$, which is not the case.

Solution for Exercise 5.14:

$$\mathsf{suc}(x) = y \quad \leftrightarrow \quad \exists v[x + v = y \wedge \forall zw(v = z + w \to (v = z \vee v = w))]$$

Another solution is to define first the ordering $\leq$ from addition as

$$x \leq y \quad \leftrightarrow \quad \exists z\, x + z = y$$

and then use the formula defining successor from ordering in Exercise 5.15.

Solution for Exercise 5.15:

$$\mathsf{suc}(x) = y \quad \leftrightarrow \quad (x \leq y \wedge \neg\exists z(x \leq z \wedge z \leq y \wedge x \neq z \wedge z \neq y))$$

Solution for Exercise 5.22: Of course, it can be changed. But then the meaning of the so changed formula won't be the same as that of $\mathsf{upvar}(x, n)$. The meaning of $\mathsf{upvar}(x, n)$ changes even if we replace $\exists si$ with $\forall si$ in a thoughtful way, i.e., let $\mathsf{upvar}'(x, n)$ denote $\forall si(\mathsf{deriv}(s, i) \wedge s_i = x \ \to \ \forall j \leq i ...)$. Then $\mathsf{deriv}'(x, n)$ will be true iff $\neg\mathsf{Fm}(x)$. Indeed, if $\mathsf{Fm}(x)$ and $n \in \omega$, then there is a derivation of x in which the first member is $=vnvn$ (in fact, any derivation

for x remains a derivation if we insert arbitrary derivations of other formulas into it). On the other hand, if $\neg \mathsf{Fm}(x)$ and $n \in \omega$, then there is no derivation s for x, and therefore any formula of form $\forall si(\mathsf{deriv}(s,i) \wedge s_i = x \to ...)$ is true.

Solution for Exercise 5.25: Let

$$U_0(x) = \exists yzR, \quad U_0(y) = \exists x(x = y \wedge U_0(x)), \quad U_0(z) = \exists x(x = z \wedge U_0(x)),$$
$$U_1(x) = \exists y(y = x \wedge U_1(y)), \quad U_1(y) = \exists xzR, \quad U_1(z) = \exists y(y = z \wedge U_1(y)),$$
$$U_2(x) = \exists z(z = x \wedge U_2(z)), \quad U_2(y) = \exists z(z = y \wedge U_2(z)), \quad U_2(z) = \exists xyR.$$

Then it is easy to check that for all $i < 3$ we have

$$\mathsf{mn}(U_i(x)) = \{\langle a, b, c\rangle : a \in U_0\},$$
$$\mathsf{mn}(U_i(y)) = \{\langle a, b, c\rangle : b \in U_0\},$$
$$\mathsf{mn}(U_i(z)) = \{\langle a, b, c\rangle : c \in U_0\}.$$

Let

$$\varphi(s) = s(x, y), \quad \varphi(\overline{s}) = s(y, x),$$
$$\varphi(\mathsf{id}_i) = U_i(x) \wedge U_i(y) \wedge x = y, \quad \varphi(\mathsf{di}_i) = U_i(x) \wedge U_i(y) \wedge x \neq y,$$
$$\varphi(U_i \times U_j) = U_i(x) \wedge U_j(y) \quad \text{for} \quad i \neq j.$$

For a formula ψ having at most x, y as free variables we define

$$\psi((x, z)) = \exists y(y = z \wedge \psi), \quad \psi((y, z)) = \exists x(x = y \wedge \psi((x, z))).$$

Then it is easy to check that

$$\mathsf{mn}(\psi((x, z))) = \{\langle a, b, c\rangle : \langle a, c, o\rangle \in \mathsf{mn}(\psi) \text{ for any } o\},$$
$$\mathsf{mn}(\psi((y, z))) = \{\langle a, b, c\rangle : \langle o, b, c\rangle \in \mathsf{mn}(\psi) \text{ for any } o\},$$
$$\mathsf{mn}(\varphi(e)) = \{\langle a, b, c\rangle : \langle a, b\rangle \in e\} \quad \text{for all } e \in \bigcup\{\mathsf{Rel}_{ij} : i, j < 2\}.$$

Now

$X(ijk, e)$ $\quad =$, by definition
$\{\langle a, b, c\rangle \in U_i \times U_j \times U_k : \langle a, b\rangle \in e_{01}, \langle b, c\rangle \in e_{12}, \langle a, c\rangle \in e_{02}\} =$, by the above
$\mathsf{mn}(U_i(x)) \cap \mathsf{mn}(U_j(y)) \cap \mathsf{mn}(U_k(z)) \cap$
$\mathsf{mn}(\varphi(e_{01})) \cap \mathsf{mn}(\varphi(e_{12})((y, z))) \cap \mathsf{mn}(\varphi(e_{02})((x, z))) =$
$\mathsf{mn}(U_i(x) \wedge U_j(y) \wedge U_k(z) \wedge \varphi(e_{01}) \wedge \varphi(e_{12})((y, z)) \wedge \varphi(e_{02})((x, z))).$

Acknowledgements

We thank our students, colleagues and friends for inspiration on the lectures and the discussions afterwards, and for pointing out errors, typos in the Lecture Notes. Special thanks go to Ka Yue Cheng, Larion Garaczi, Dávid András Imre and Péter Juhász. Research connected to this lecture was supported by OTKA grant No 81188.

References

[1] Andréka, H., Comer, S. D., Madarász, J. X., Németi, I., and Sayed-Ahmed, T., Epimorphisms in cylindric algebras and definability in finite variable logic. Algebra Universalis 61,3-4 (2009), 261-282. Springer-Verlag, 1985.

[2] Andréka, H., van Benthem, J., and Németi, I., Modal languages and bounded fragments of predicate logic. Journal of Philosophical Logic 27 (1998), 217-274.

[3] Barwise, J., and Feferman, S., (eds.) *Model-Theoretic Logics*. Springer-Verlag, 1985.

[4] van Benthem, J., The logical study of science. Synthese 51 (1982), 431-472.

[5] van Benthem, J., and Pearce, D., A mathematical characterization of interpretation between theories. Studia Logica 43,3 (1984), 295-303.

[6] Burstall, R., and Goguen, J., Putting theories together to make specifications. In: Proc. IJCAI'77 (Proceedings of the 5th International Joint Conference on Artificial Intelligence, Vol 2, pp.1045-1058. Morgan Kaufmann Puboishers Inc. San Francisco, CA, USA, 1977.

[7] Cegielski, P., Matiyasevich, Y., Richard, D., Definability and decidability issues in extensions of the integers with the divisibility predicate. The Journal of Symbolic Logic 61,2 (1996), 515-540.

[8] Chang, C. C., and Keisler, H. J., *Model theory*. North-Holland, 1973.

[9] Cooper, S. B., *Computability Theory*, Chapman & Hall, 2004.

[10] Enderton, H. B., *A mathematical introduction to logic*, Academic Press, 1972.

[11] Feferman, S., and Vaught, R., The first order properties of algebraic systems. Fundamenta Mathematicae 47 (1959), 57-103.

[12] Friedman, H., On foundational thinking 1. FOM (Foundations of Mathematics) Posting, Archives www.cs.nyu.edu, January 20, 2004.

[13] Henkin, L., Monk, J. D., and Tarski, A., Cylindric Algebras. Part I. North-Holland, 1971.

[14] Hodges, W., Model Theory. Cambridge University Press, 1993.

[15] Hodkinson, I., Finite variable logics. Bull. Europ. Assoc. Theor. Comp. Sci. 51 (1993) 111-140. With addendum, volume 52. `http://www.doc.ic.ac.uk/~imh/papers/fvl_revised.pdf`

[16] Kaye, R., Wong, T. L., On interpretations of arithmetic and set theory. Notre Dame Journal of Formal Logic 48,4 (2007), 497-510.

[17] Madarász, J. X., Logic and relativity (in the light of definability theory). PhD Dissertation, Eötvös Loránd University, 2002. http://www.math-inst.hu/pub/algebraic-logic/diszi.pdf

[18] Makkai, M., First order logic with dependent sorts, with applications to category theory. Preliminary version Nov 6, 1995, McGill University. http://www.math.mcgill.ca/makkai/folds/foldsinpdf/FOLDS.pdf

[19] Mostowski, A., Craig's interpolation theorem in some extended systems of logic. In: Logic, methodology and philosophy of science III, North-Holland, 1968. pp.87-103.

[20] Tarski, A., and Givant, S. R., A formalization of set theory without variables. American Mathematical Society Vol 41., 1987. 318pp.

Alfréd Rényi Institute of Mathematics, Hungarian Academy of Sciences
Budapest, Reáltanoda st. 13-15, H-1053 Hungary
andreka.hajnal@renyi.mta.hu, nemeti.istvan@renyi.mta.hu

Index