Successor Levels of the Jensen Hierarchy*

Gunter Fuchs
Westfälische Wilhelms-Universität Münster
Germany

April 25, 2008

Abstract

I prove that there is a recursive function T that does the following: Let X be transitive and rud closed, and let X' be the closure of $X \cup \{X\}$ under rud functions. Given a Σ_0 formula $\phi(x)$ and a code c for a rud function f, $T(\phi, c, \vec{x})$ is a Σ_ω formula such that for any $\vec{a} \in X$, $X' \models \phi[f(\vec{a})]$ iff $X \models T(\phi, c, \vec{x})[\vec{a}]$. I make this precise and show relativized versions of this. As an application, I prove that under certain conditions, if Y is the Σ_ω extender ultrapower of X with respect to some extender F that also is an extender on X', then the closure of $Y \cup \{Y\}$ under rud functions is the Σ_0 extender ultrapower of X' with respect to F, and the ultrapower embeddings agree on X.

1 Introduction

There are many situations in inner model theory where it is necessary to express a definition over a successor level of a relativized Jensen-hierarchy by a (more complex) definition over the predecessor level, in a uniform way. The author came across this problem when trying to establish a one-to-one correspondence between (pre-)mice in the setting of [MS94] and those in the Friedman-Jensen style (cf. [Jen97], [Zem02]). This was done in the author's dissertation [Fuc03]. Stripping away many details and complications, the situation in which the tool presented here is called for is as follows.

Let's assume the two "corresponding" structures $\bar{M} = \langle \mathrm{J}^A_{\bar{\alpha}}, B\rangle$ (where, for simplicity, $A, B \subseteq \mathrm{J}^A_{\bar{\alpha}}$) and $\bar{M}' = F(\bar{M}) = \langle \mathrm{J}^{A'}_{\bar{\alpha}'}, B'\rangle$ have been defined already. Again, suppose that $A', B' \subseteq \mathrm{J}^{A'}_{\bar{\alpha}'}$. The function F is defined by recursion, and,

*MSC 2000: 03E45. *Keywords:* Rudimentary closure, extenders, ultrapowers

say we are in a successor case of the definition, where, setting $M = \langle \mathrm{J}^A_{\bar\alpha+1}, B\rangle$, we have that $M' := F(M) =$ "$\bar M' + 1$" $= \langle \mathrm{J}^{A'}_{\bar\alpha'+1}, B'\rangle$. I want the structures to have "the same fine structural properties". For example, I want the projecta to be the same, and so on. What this amounts to is an analysis of Σ_1-definability over these structures. Suppose we have a translation $t_{\bar M}$ of Σ_ω formulae $\psi(\vec x)$ such that

$$\bar M \models \psi[\vec a] \iff \bar M' \models t_{\bar M}(\psi)[\vec a].$$

In order to expand this translation function further, suppose $\phi(x)$ to be a Σ_0 formula, and let c be a rudimentary term for an element of the A-rudimentary closure of a set.[1] So let c code $f(-, |\bar M|)$, and let $\vec a \in \bar M$. I would like to be able to argue as follows:

$$\begin{aligned} & M \models \phi[f(\vec a, |\bar M|)] \\ \iff & \bar M \models T(\phi, c, \vec x)[\vec a] \\ \iff & \bar M' \models t_{\bar M}(T(\phi, c, \vec x))[\vec a] \\ \iff & M' \models t_{\bar M}(T(\phi, c, \vec x))_{\bar M'}[\vec a]. \end{aligned}$$

Expanding this method to, say, Σ_1 formulae amounts to quantifying over rudimentary terms. So let $\mathfrak{T}(\dot A)$ denote the set of A-rudimentary terms, and let $\phi(y, x)$ be a Σ_0 formula. One can then proceed as follows:

$$\begin{aligned} & M \models \exists y \quad \phi[y, f(\vec a, |\bar M|)] \\ \iff & \exists d \in \mathfrak{T}(\dot A) \exists \vec b \in \bar M \quad M \models \phi[f_d(\vec b, |\bar M|), f(\vec a, |\bar M|)] \\ \iff & \exists d \in \mathfrak{T}(\dot A) \quad \bar M \models (\exists \vec y \quad T(\phi, x, c, \vec x, y, d, \vec y))[({}^{\vec x}/{}_{\vec a})] \\ \iff & \exists d \in \mathfrak{T}(\dot A) \quad \bar M' \models t_{\bar M}(\exists \vec y \quad T(\phi, x, c, \vec x, y, d, \vec y))[\vec a] \\ \iff & M' \models \exists d \in \mathfrak{T}(\dot A) \quad (\bar M' \models t_{\bar M}(\exists \vec y \quad T(\phi, x, c, \vec x, y, d, \vec y))[\vec a]). \end{aligned}$$

I wrote f_d to denote the A-rud function which is coded by d. The presentation is intentionally very sketchy; e.g., it is unclear which parameters are to be allowed in the formulae that are translated. A detailed exposition will be given in the forthcoming article [Fuc].

The basic idea of how to define the function T that enables me to argue as indicated above stems from the first pages of the foundational article [Jen72], and hence from the very beginnings of fine structure theory. The information gained from [Dev84, Chapter IV, p. 246, Lemma 1.18] is not sufficient for the intended applications of the translation procedure; what is needed is a recursive and uniform way to translate formulae in the way I described.

[1] I am a bit sloppy here: Every element y of the A-rud closure of a set X has the form $y = f(\vec a, X)$, where f is rud in A and $\vec a \in X$. The terms I am talking about are basically names for the functions $\vec x \mapsto f(\vec x, X)$.

The article is organized as follows. In the first section, in order to be able to state the results precisely, I introduce a coding of rudimentary functions. Which coding is chosen is probably not very important, as long as it is reasonably simple. For example, since the rudimentary functions have a finite basis, one could use terms in the language consisting of function symbols representing the basic functions. Or one could view a rudimentary function as a finite sequence of defining schemes. The coding chosen here is pretty much equivalent to the latter approach but has notational advantages. Compared to the former coding, it seems to facilitate inductive arguments. Using this coding, I prove a substitution lemma for Σ_0 formulae.

The second section introduces rudimentary terms, which can be viewed as codes for the elements of successor levels of the Jensen hierarchy. Subsequently, the translation function is defined.

Finally, in the last section, I give an application on extender ultrapowers of successor levels, which makes use of the translation procedure of the second section.

The notation I use is quite standard. I should maybe say that I write $|M|$ for the universe of a model M (not for its cardinality), $\vec{x}$ is short for a finite list $x_1, \ldots, x_m$, $\langle\vec{x}\rangle$ is the m-tuple, and $\prec\vec{\alpha}\succ$ is the value of $\langle\vec{\alpha}\rangle$ under the Gödel pairing function.

2 Substitution and Codes for Rudimentary Functions

In order to be able to state the results, I fix the following coding of rudimentary functions.

Definition 2.1. Let $\vec{\dot{A}} = \dot{A}_1, \ldots, \dot{A}_l$ be a list of predicate symbols. Since I shall be working with transitive structures that are closed under ordered pairs, I shall once and for all restrict to *unary predicate symbols*. The set $\mathfrak{C}(\vec{\dot{A}})$ of codes for functions rudimentary in $\vec{\dot{A}}$ is defined by the following clauses.

(a) For all $n \in \omega \setminus \{0\}$ and $k, l < n$, the following symbols are codes for an n-ary function rudimentary in $\vec{\dot{A}}$:
$$\pi^n_k, p^n_{k,l}, \delta^n_{k,l}.$$

(b) The symbol $f_{\dot{A}_k}$ is a code for a 1-ary function rudimentary in $\vec{\dot{A}}$ $(1 \leq k \leq l)$.

(c) If f is a code for an n-ary function rudimentary in $\vec{\dot{A}}$, then so is $u^n[f]$.

(d) If h is a code for an m-ary function rudimentary in $\vec{A}$ and $h_0, \ldots, h_{m-1}$ are codes for n-ary functions rudimentary in $\vec{A}$, then $h \circ (h_0, \ldots, h_{m-1})$ is a code for an n-ary function rudimentary in $\vec{A}$ $(m, n \geq 1)$.

Now I am going to define the interpretations of codes for functions rudimentary in $\vec{A}$:

Fix sets (or classes) $\vec{A} := A_1, \ldots, A_l$. Given a code t for an n-ary function in $\mathfrak{C}(\vec{A})$, I define its interpretation, $\mathtt{val}^{\vec{A}}[t] : \mathrm{V}^n \longrightarrow \mathrm{V}$ by recursion on t as follows.[2]

(a) Let $n \in \omega \setminus \{0\}$, $k, l < n$. Then set:

(1) $\mathtt{val}^{\vec{A}}[\pi^n_k](a_0, \ldots, a_{n-1}) = a_k$.

(2) $\mathtt{val}^{\vec{A}}[p^n_{k,l}](a_0, \ldots, a_{n-1}) = \{a_k, a_l\}$.

(3) $\mathtt{val}^{\vec{A}}[\delta^n_{k,l}](a_0, \ldots, a_{n-1}) = a_k \setminus a_l$.

(b) $\mathtt{val}^{\vec{A}}[f_{\dot{A}_k}](a) = A_k \cap a$ $(1 \leq k \leq l)$.

(c) Let f be a code for an n-ary function rudimentary in $\vec{A}$ for which $\mathtt{val}^{\vec{A}}[f]$ has been defined already. Then set:

$$\mathtt{val}^{\vec{A}}[u^n[f]](a_0, \ldots, a_{n-1}) = \bigcup_{b \in a_0} \mathtt{val}^{\vec{A}}[f](b, a_1, \ldots, a_{n-1}).$$

(d) Let h be a code for an m-ary function rudimentary in $\vec{A}$, and let $h_0, \ldots, h_{m-1}$ be codes for n-ary functions rudimentary in $\vec{A}$, such that $\mathtt{val}^{\vec{A}}[h]$ and $\mathtt{val}^{\vec{A}}[h_0], \ldots, \mathtt{val}^{\vec{A}}[h_m]$ have already been defined. Then, for $\vec{a} = a_0, \ldots, a_{n-1}$, set:

$$\mathtt{val}^{\vec{A}}[h \circ (h_0, \ldots, h_{m-1})](\vec{a}) = \mathtt{val}^{\vec{A}}[h](\mathtt{val}^{\vec{A}}[h_0](\vec{a}), \ldots, \mathtt{val}^{\vec{A}}[h_{m-1}](\vec{a})).$$

I shall freely confuse codes for rudimentary functions with their Gödel numbers. The following result is a refinement of [Jen72, Lemma 1.2., p. 235] (see also [Dev84, Lemmata IV.1.17,18]), which says that rudimentary functions are simple, meaning that the substitution of a rudimentary function in a Σ_0 relation again is a Σ_0 relation. What matters here, though, is that if I am given a Σ_0-formula and codes for the rudimentary functions to be substituted, then I can *effectively*

[2]The relation over which the recursive definition is formulated is the immediate subcode relation. For further details, see p. 5.

compute a Σ_0 formula that defines the relation which is the result of substituting the functions into the relation defined by the original Σ_0 formula.

For the rest of the paper, fix a language $\mathcal{L}^*$ which is the language of set theory with additional predicate symbols $\vec{\dot{A}} := \dot{A}_1, \ldots, \dot{A}_p$ and $\vec{\dot{B}} := \dot{B}_1, \ldots, \dot{B}_q$.

Definition 2.2. Let ϕ be an $\mathcal{L}^*$-formula. Then a variable v which is not bound in ϕ is *basic* in ϕ if ϕ has no subformula of the form $\dot{A}_k(v)$ $(1 \leq k \leq p)$ or $\dot{B}_l(v)$ $(1 \leq l \leq q)$.

An *assignment in X* is a finite function whose domain is a set of variables and whose range is contained in X. If b is an assignment in X, v is a variable and $a \in X$, then $b({}^{v}/_{a})$ is the assignment with domain $\mathrm{dom}(b) \cup \{v\}$ (note that $v \in \mathrm{dom}(b)$ is allowed) which coincides with b at all variables, with the possible exception of v, which is mapped to a. An *assignment for ϕ in X* is an assignment in X whose domain contains the set of free variables of ϕ.

Lemma 2.3. *There is a recursive function* Sub′ *(only depending on $\mathcal{L}^*$) with the following property.*

Let $\phi = \phi(v_0, \ldots, v_{n-1})$ be a Σ_0 formula in $\mathcal{L}^*$. Fix interpretations A_k, B_l of $\dot{A}_k$, $\dot{B}_l$, respectively. Let $i^* < n$ be such that v_{i^*} is basic in ϕ. Let $c^* \in \mathfrak{C}(\vec{A})$ be a code for an m-ary function rudimentary in $\vec{A}$, and let $x_0, \ldots, x_{m-1}$ be a list of variables not containing the variable v_{i^*}, nor any bound variable of ϕ.

Then $\psi := \mathrm{Sub}'(\phi, v_{i^*}, c^*, \langle\vec{x}\rangle)$ is again a Σ_0 formula in $\mathcal{L}^*$ whose free variables are in $\{\vec{x}, \vec{v}\} \setminus \{v_{i^*}\}$, and which has the property that for any assignment b for ψ in V*,*

$$\langle \mathrm{V}, \vec{A}, \vec{B}\rangle \models \phi[b({}^{v_{i^*}}/_{\mathtt{val}^{\vec{A}}[c^*](b(\vec{x}))})] \iff \langle \mathrm{V}, \vec{A}, \vec{B}\rangle \models \psi[b].$$

Moreover,

(#) *Every variable that is basic in ϕ is basic in ψ, as well.*

Proof. I am going to define a function with the desired properties by recursion on a set-like well founded relation which I introduce below.

Firstly, it's clear what is meant by saying that a rudimentary code c is an *immediate subcode* of another rudimentary code d: Either $d = u^l[c]$ for an $l \in \omega$, or $d = h \circ (h_0, \ldots, h_{l-1})$, where $c \in \{h, h_0, \ldots, h_{l-1}\}$.

Call $a = \langle \phi, v, c, \langle\vec{x}\rangle\rangle$ a *permissible argument*, if, according to the statement of the lemma, a should be in the domain of Sub′. For a permissible argument a, I say:

(A) a is of type (A), if a has the form $\langle u \in v, v, c, \langle\vec{x}\rangle\rangle$.

(B) a is of type (B), if a has the form $\langle \forall y \in z \quad \psi, z, c, \langle\vec{x}\rangle\rangle$, where z doesn't occur in ψ.

Now let $a_1 := \langle \phi_1, v_1, c_1, \langle \vec{x}^1 \rangle \rangle$ and $a_2 := \langle \phi_2, v_2, c_2, \langle \vec{x}^2 \rangle \rangle$ be permissible arguments. Then I set $a_1 \prec a_2$ iff one of the following possibilities holds true:

1. c_1 is an immediate subcode of c_2.
2. $c_1 = c_2$, a_1 is of type (A) or (B), and a_2 is not of of type (A) or (B).
3. $c_1 = c_2$, a_1 is not of type (A) or (B), a_2 is not of type (A) or (B), and ϕ_1 is a subformula of ϕ_2.

Now I define, by $\prec$-recursion on permissible arguments $a = \langle \psi, v, c, \langle \vec{x} \rangle \rangle$, the value $\mathrm{Sub}'(a)$. There is an additional clause to every definition, which I mention only once:

$$\text{If } v \text{ does not occur in } \psi, \text{ then } \mathrm{Sub}'(a) := \psi.$$

Also, there is a technical issue which I would like to address now and then suppress in the rest of the proof when it is not important. The problem is that if $\mathrm{Sub}'(a) = \psi'$, then ψ' will generally contain more bound variables than ψ contained. Suppose now that $a \prec b$, and that $\mathrm{Sub}'(a)$ has been defined. Since $\mathrm{Sub}'(a)$ has been defined "out of context", i.e., without knowing b, it may occur that some of the newly introduced bound variables in ψ' are free in the formula of b, i.e., in the first component of the argument b. But in defining $\mathrm{Sub}'(b)$, I want to refer to $\mathrm{Sub}'(a)$. There are several ways to deal with this problem: One could rename the bound variables of $\mathrm{Sub}'(a)$ when using it in the definition of $\mathrm{Sub}'(b)$. The cleaner approach seems to be to define a preliminary function $\overline{\mathrm{Sub}}'$ which takes an additional argument b, a finite set of "forbidden" variables, a "black list". For a permissible argument c, one defines $\chi = \overline{\mathrm{Sub}}'(c, \mathsf{b})$ by recursion in such a way that in the end, one may set $\mathrm{Sub}'(c) = \overline{\mathrm{Sub}}'(c, \emptyset)$, and that in addition, none of the variables in b occur in χ as a bound variable. The variables occurring in c will implicitly be treated as though they were on the black list, so the definition $\mathrm{Sub}'(c) = \overline{\mathrm{Sub}}'(c, \emptyset)$ is right. Whenever I pick a bound variable in the course of the proof, this choice can be made effectively by presuming an enumeration of the variables, and then choosing the one with the least index among those that are not on the black list b, which is sometimes suppressed in the notation. The ordering on pairs $\langle c, \mathsf{b} \rangle$, where c is a permissible argument and b is a black list, along which the recursive definition shall proceed, is induced by $\prec$ by simply ignoring the second component:

$$\langle c, \mathsf{b} \rangle \prec' \langle c', \mathsf{b}' \rangle \iff c \prec c'.$$

At sensitive places in the construction, I'm going to write down the details explicitly.

Let's begin with the recursive definition now.

Case 1: c is a primitive code (i.e., c has no immediate subcode).

Case 1.1: a is of type (A) or (B).

Then, depending on the type of a, I stipulate:

(A) Let $\psi \equiv (w \in v)$. If w and v are not identical, then

$$\begin{array}{lcl} \mathrm{Sub}'(w \in v, v, \pi^n_i, \langle \vec{x} \rangle) & := & w \in x_i, \\ \mathrm{Sub}'(w \in v, v, p^n_{k,l}, \langle \vec{x} \rangle) & := & (w = x_k \vee w = x_l), \\ \mathrm{Sub}'(w \in v, v, \delta^n_{k,l}, \langle \vec{x} \rangle) & := & (w \in x_k \wedge \neg(w \in x_l)), \\ \mathrm{Sub}'(w \in v, v, f_{\dot{A}_k}, x) & := & \exists z \in x \quad (z = w \wedge \dot{A}_k(z)). \end{array}$$

If w and v are identical, then $\mathrm{Sub}'(a) := (x_0 \in x_0)$.

(B)

$$\begin{array}{lcl} \mathrm{Sub}'(\forall y \in v \quad \phi, v, \pi^n_i, \langle \vec{x} \rangle) & := & \forall y \in x_i \quad \phi, \\ \mathrm{Sub}'(\forall y \in v \quad \phi, v, p^n_{k,l}, \langle \vec{x} \rangle) & := & (\phi({}^y/{}_{x_k}) \wedge \phi({}^y/{}_{x_l})), \\ \mathrm{Sub}'(\forall y \in v \quad \phi, v, \delta^n_{k,l}, \langle \vec{x} \rangle) & := & \forall y \in x_k \quad (y \notin x_l \longrightarrow \phi), \\ \mathrm{Sub}'(\forall y \in v \quad \phi, v, f_{\dot{A}_k}, x) & := & \forall y \in x \quad (\dot{A}_k(y) \longrightarrow \phi). \end{array}$$

The definition in the fourth case of (A) may seem unnecessarily complicated, but note that just taking the formula $\dot{A}_k(w) \wedge w \in x$ as the definition won't do, since this would make $\dot{A}_k(w)$ a subformula of $\mathrm{Sub}'(w \in v, v, f_{\dot{A}_k}, x)$, which is forbidden by (#). Note also that I introduced a new bound variable in this case. This is a place where I really use the black list that's dragged along as an additional argument. So, to be explicit, I define

$$\overline{\mathrm{Sub}}'(w \in v, v, f_{\dot{A}_k}, x, \mathsf{b}) \ := \ \exists z \in x \quad (z = w \wedge \dot{A}_k(z)),$$

where z is the variable with least index (as a short form for this, I'll just say "the least variable" in the future) that's not on the black list b and that doesn't occur in $\{v, w, x\}$.

In case w and v are identical, I had to define $\mathrm{Sub}'(w \in v, v, c, \vec{x})$ to be something like $(x_0 \in x_0)$, because v is not allowed to be a free variable in the resulting formula anymore.

Case 1.2: a is not of type (A) or (B), and ψ is atomic.

It cannot be the case that $\psi \equiv \dot{A}_k(v)$ or $\psi \equiv \dot{B}_l(v)$, as a is a permissible argument. So only the following subcases may occur:

Case 1.2.1: $\psi \equiv w = v$.

If w and v are identical, i.e., if $\psi \equiv v = v$, then I set: $\mathrm{Sub}'(a) := x_0 = x_0$. The reason is, again, that v is not allowed to occur as a free variable in the resulting formula. If w and v are distinct, then I define $\mathrm{Sub}'(w = v, v, c, \langle\vec{x}\rangle)$ to be the following formula:

$$(\forall z \in w \quad \mathrm{Sub}'(\underbrace{z \in v, v, c, \langle\vec{x}\rangle}_{\text{type (A)}})) \wedge \mathrm{Sub}'(\underbrace{\forall z \in v \quad z \in w, v, c, \langle\vec{x}\rangle}_{\text{type (B)}}).$$

To be more precise, I'll give the real definition of $\overline{\mathrm{Sub}}'(w = v, v, c, \langle\vec{x}\rangle, \mathsf{b})$ in this case:

$$(\forall z \in w \quad \overline{\mathrm{Sub}}'(\underbrace{z \in v, v, c, \langle\vec{x}\rangle, \mathsf{b} \cup \{w\}}_{\text{type (A)}})) \wedge \overline{\mathrm{Sub}}'(\underbrace{\forall z \in v \quad z \in w, v, c, \langle\vec{x}\rangle, \mathsf{b}}_{\text{type (B)}}),$$

where z is the least variable that does not occur in $\mathsf{b} \cup \{v, w, \vec{x}\}$. The case that the other variable is to be substituted is symmetric.

Case 1.2.2: $\psi \equiv v \in w$.

Since a is not of type (A), the following definition will do:

$$\mathrm{Sub}'(v \in w, v, c, \langle\vec{x}\rangle) := \exists z \in w \quad \underbrace{\mathrm{Sub}'(z = v, v, c, \langle\vec{x}\rangle)}_{\text{defined as in case 1.2.1}}.$$

Again, z has to be picked according to the suppressed bookkeeping system: So I define $\overline{\mathrm{Sub}}'(v \in w, v, c, \langle\vec{x}\rangle, \mathsf{b})$ to be the formula

$$\exists z \in w \quad \overline{\mathrm{Sub}}'(z = v, v, c, \langle\vec{x}\rangle, \mathsf{b} \cup \{w\}),$$

where z is the least variable which does not occur in $\mathsf{b} \cup \{v, w, \vec{x}\}$.

Case 1.3: $\psi \equiv \neg\phi'$ or $\phi \equiv \phi_1 \circ \phi_2$, where $\circ = \wedge, \vee, \rightarrow, \leftrightarrow$.

In this case I make the obvious definitions:

$$\begin{aligned} \mathrm{Sub}'(\neg\phi', x, c, \langle\vec{x}\rangle) &:= \neg\mathrm{Sub}'(\phi', x, c, \langle\vec{x}\rangle) \\ \mathrm{Sub}'(\phi_1 \circ \phi_2, x, c, \langle\vec{x}\rangle) &:= \mathrm{Sub}'(\phi_1, x, c, \langle\vec{x}\rangle) \circ \mathrm{Sub}'(\phi_2, x, c, \langle\vec{x}\rangle). \end{aligned}$$

Case 1.4: $\psi \equiv \forall v \in w \quad \phi$, where $\phi = \phi(v, w, \vec{z})$ and a is neither of type (A) nor of type (B).

If w is to be substituted, then it also appears in ϕ since otherwise the argument would be of type (B). So in this case, I set:

$$\mathrm{Sub}'(\forall v \in w \quad \phi, w, c, \langle \vec{x} \rangle) := \mathrm{Sub}'(\underbrace{\forall v \in w \quad \mathrm{Sub}'(\phi, w, c, \langle \vec{x} \rangle), w, c, \langle \vec{x} \rangle}_{a'}).$$

Note that w does not occur in $\mathrm{Sub}'(\phi, w, c, \langle \vec{x} \rangle)$. Hence, the permissible argument a' is of type (B), and hence, $\mathrm{Sub}'(a')$ is already defined. Really, one should add v to the black list in the nested function call to $\overline{\mathrm{Sub}}'$ here.

If it isn't w that is to be substituted, I make the following definition:

$$\mathrm{Sub}'(\forall v \in w \quad \phi, z_i, c, \langle \vec{x} \rangle) := \forall v \in w \quad \mathrm{Sub}'(\phi, z_i, c, \langle \vec{x} \rangle).$$

Here, one really should add v and w to the black list in the function call to $\overline{\mathrm{Sub}}'(\ldots)$, in order to insure that these variables aren't introduced as new bound variables there.

Case 1.5: $\psi \equiv \exists w \in v \quad \phi$.

This case can be reduced to the case $\neg(\forall w \in v \quad \neg\phi)$ as usual.

Case 2: $c = u^n[g]$.

Case 2.1: The argument is of type (A) or (B).

If the argument is of type (A), I set:

$$\mathrm{Sub}'(w \in v, v, u^n[g], \langle \vec{x} \rangle) := \exists z \in x_0 \quad \mathrm{Sub}'(w \in z', z', g, \langle z, x_1, \ldots, x_{n-1} \rangle),$$

where z and z' are new variables and $\vec{x} = x_0, x_1, \ldots, x_{n-1}$. More precisely, $\overline{\mathrm{Sub}}'(w \in v, v, u^n[g], \langle \vec{x} \rangle, \mathsf{b})$ is defined to be the following formula:

$$\exists z \in x_0 \quad \overline{\mathrm{Sub}}'(w \in z', z', g, \langle z, x_1, \ldots, x_{n-1} \rangle, \mathsf{b} \cup \{x_0\}),$$

where z and z' are the next two variables that are not in $\mathsf{b} \cup \{v, w, \vec{x}\}$.

Otherwise, if the argument is of type (B), I define:

$$\begin{aligned} &\mathrm{Sub}'(\forall w \in v \quad \phi, v, u^n[g], \langle \vec{x} \rangle) := \\ &\qquad\qquad \forall z \in x_0 \quad \mathrm{Sub}'(\forall w \in z' \quad \phi, z', g, \langle z, x_1, \ldots, x_{n-1} \rangle). \end{aligned}$$

Again, z and z' have to be picked relative to the suppressed black list, and x_0 has to be added to the black list in the function call to $\overline{\mathrm{Sub}}'$ on the right hand side.

The other cases 2.2-2.5 can be dealt with like the cases 1.2-1.5; there, it didn't matter there that the substituted code was primitive.

Case 3: $c = h \circ (h_0, \ldots, h_{m-1})$.

Choose new variables z_0, ..., z_{m-1} (relative to the black list) and then define $\mathrm{Sub}'(\psi, v, c, \langle\vec{x}\rangle)$ to be the formula ψ', where

$$\psi' := \mathrm{Sub}'(\mathrm{Sub}'(\phi, v, h, \langle\vec{z}\rangle), z_0, h_0, \langle\vec{x}\rangle, \ldots, z_{m-1}, h_{m-1}, \langle\vec{x}\rangle).$$

I used a suggestive yet sloppy notation here. What's really meant is the following:

$$\psi' := \mathrm{Sub}'(\mathrm{Sub}'(\ldots \mathrm{Sub}'(\mathrm{Sub}'(\mathrm{Sub}'(\psi, v, h, \vec{z}), z_0, h_0, \langle\vec{x}\rangle), \\ , z_1, h_1, \langle\vec{x}\rangle) \ldots), z_{m-1}, h_{m-1}, \langle\vec{x}\rangle).$$

In the innermost function call $\mathrm{Sub}'(\psi, v, h, \vec{z})$, it really has to be insured that none of the variables $\vec{x}$ are added as bound variables, by adding them to the black list. After this, the other function calls are possible. So the exact definition would be that $\overline{\mathrm{Sub}}'(\psi, v, c, \langle\vec{x}\rangle, \mathsf{b})$ is the following formula:

$$\overline{\mathrm{Sub}}'(\overline{\mathrm{Sub}}'(\ldots \overline{\mathrm{Sub}}'(\overline{\mathrm{Sub}}'(\overline{\mathrm{Sub}}'(\psi, v, h, \langle\vec{z}\rangle, \mathsf{b} \cup \{\vec{x}\}), z_0, h_0, \langle\vec{x}\rangle, \mathsf{b}), \\ , z_1, h_1, \langle\vec{x}\rangle, \mathsf{b}) \ldots), z_{m-1}, h_{m-1}, \langle\vec{x}\rangle, \mathsf{b}).$$

It is obvious that all arguments appearing in this formula are $\prec a$, as the codes substituted are immediate subcodes of c.

This completes the definition of Sub'. It is easy to verify that it has the desired properties. $\square$

The function Sub' allows for the substitution of one variable. Now I am going to define a function that makes multiple simultaneous substitutions possible.

Lemma 2.4. *There is a recursive function* Sub *such that the following holds.*

Let $\phi = \phi(v_0, \ldots, v_{n-1})$ be a Σ_0 formula $\mathcal{L}^*$. Fix interpretations A_k, B_l of $\dot{A}_k$ and $\dot{B}_l$, respectively.

Let $a = \{i_0, \ldots, i_{m-1}\}$ be an m-element subset of n, such that for $j < m$, v_{i_j} is basic in ϕ. Moreover, for $j < m$, let $c_j \in \mathfrak{C}(\vec{\dot{A}})$ be a code for an n_j-ary function, and $\vec{x}^j := x^j_0, \ldots, x^j_{n_j-1}$ a list of variables not bound in ϕ.

Then $\psi := \mathrm{Sub}(\phi, v_{i_0}, c_0, \langle\vec{x}^0\rangle, \ldots, v_{i_{m-1}}, c_{m-1}, \langle\vec{x}^{m-1}\rangle)$ is a Σ_0 formula in $\mathcal{L}^*$ that has the following property:

If $\vec{w} = w_0, \ldots, w_{m'-1}$ is an enumeration of $\{v_k \mid k \in n \setminus a\}$, then the set of free variables of ψ is contained in $\{\vec{w}, \vec{x}^0, \ldots, \vec{x}^{m-1}\}$ – this is a list with possible repetitions. Let b be an assignment of these variables. Then we have:

$$\langle \mathrm{V}, \vec{A}, \vec{B} \rangle \models \phi[({}^{v_{i_0}}/_{\mathtt{val}^{\vec{A}}[c_0](b(\vec{x}^0))}), \ldots, ({}^{v_{i_{m-1}}}/_{\mathtt{val}^{\vec{A}}[c_{m-1}](b(\vec{x}^{m-1}))}), ({}^{\vec{w}}/_{b(\vec{w})})]$$
$$\Longleftrightarrow$$
$$\langle \mathrm{V}, \vec{A}, \vec{B} \rangle \models \psi[b].$$

Proof. Let $\langle \phi, v_{i_0}, c_0, \langle \vec{x}^0 \rangle, \ldots, v_{i_{m-1}}, c_{m-1}, \langle \vec{x}^{m-1} \rangle \rangle$ be a suitable argument. Iterated application of the function Sub$'$ from the previous lemma will achieve the simultaneous replacement of several variables. The following procedure describes how to compute $\mathrm{Sub}(\phi, v_{i_0}, c_0, \langle \vec{x}^0 \rangle, \ldots, v_{i_{m-1}}, c_{m-1}, \langle \vec{x}^{m-1} \rangle)$. First, I will define a sequence $\langle \phi_0, \ldots, \phi_{m-1} \rangle$ of formulae by induction.

Let $\vec{y}^0 = y^0_0, \ldots, y^0_{n_0-1}$ be the least n_0 variables that don't occur in ϕ or in $\vec{x}^0, \ldots, \vec{x}^{m-1}$. Set:

$$\phi_0 = \mathrm{Sub}(\phi, v_{i_0}, c_0, \langle \vec{y}^0 \rangle).$$

Analogously, if $l+1 < m$ and ϕ_l has already been defined, then let $\vec{y}^{l+1}$ be the least n_{l+1} variables which don't occur in ϕ_l or in $\vec{x}^0, \ldots, \vec{x}^{m-1}$, and set:

$$\phi_{l+1} = \mathrm{Sub}(\phi_l, v_{i_{l+1}}, c_{l+1}, \langle \vec{y}^{l+1} \rangle).$$

That the arguments occurring here are permissible is guaranteed by the property (#) that Sub$'$ satisfies.

Now ϕ_{m-1} is almost as wished. All that's left to do seems to be to rename $\vec{y}^0$ by $\vec{x}^0$, etc. But it might be that some of the x^k_l are bound variables in ϕ_{m-1}. So let ϕ'_{m-1} result from ϕ_{m-1} by renaming bound variables so that this does not occur anymore. This can be done effectively: Let $\vec{w}$ list the x^k_l which are bound in ϕ_{m-1}, in increasing order. Let r be the length of this list. Let $\vec{w}'$ list the r next variables which do not occur in ϕ_{m-1} and which are different from $\vec{x}^0, \ldots, \vec{x}^{m-1}$. Let $\phi'_{m-1} = \phi_{m-1}({}^{\vec{w}}/_{\vec{w}'})$. Now I can define $\mathrm{Sub}(\phi, v_{i_0}, c_0, \langle \vec{x}^0 \rangle, \ldots, v_{i_{m-1}}, c_{m-1}, \langle \vec{x}^{m-1} \rangle)$ to be the formula

$$\phi'_{m-1}({}^{\vec{y}^0}/_{\vec{x}^0}) \cdots ({}^{\vec{y}^{m-1}}/_{\vec{x}^{m-1}}).$$

This formula has the desired properties, and the process defining it is clearly recursive. $\square$

3 The Translation Procedure and Rudimentary Terms

Now I am aiming at expressing Σ_0 definability over the rudimentary closure of some set over the set itself. To this end, I introduce rudimentary terms that represent elements of the rudimentary closure of a set. In order to avoid a possible confusion, since there are conflicting definitions in the literature, it should be pointed out that **by $\mathrm{rud}_{\vec{A}}(X)$ I mean the closure of $X \cup \{X\}$ under functions rudimentary**

in $\vec{A}$. That's what I refer to as the $\vec{A}$-rudimentary closure of X. So every element of $\mathrm{rud}_{\vec{A}}(X)$ is of the form $f(\vec{a}, X)$, where f is a function rudimentary in $\vec{A}$ and $\vec{a} \in X$. This is the motivation for the following two definitions.

Definition 3.1. Fix predicate symbols $\dot{\vec{A}}$. The set $\mathfrak{T}(\dot{\vec{A}})$ of terms rudimentary in $\dot{\vec{A}}$ is defined to consist of pairs $t = \langle c, \langle \vec{x} \rangle \rangle$, where $c \in \mathfrak{C}(\dot{\vec{A}})$ is a code for an n-ary function and $\vec{x} = \langle x_0, \ldots, x_{n-1} \rangle$ is an n-tuple, such that, for $i < n$, either x_i is a variable, or $x_i = \Phi$ for a fixed new constant symbol Φ.

The set of free variables of t, $\mathrm{Fr}(t)$ is defined to be $\{x_i \mid x_i \neq \Phi\}$.

Evaluations of rudimentary terms are now computed relative to a given interpretation of the predicate symbols and a given interpretation of a universe.

Definition 3.2. I evaluate a rudimentary term $t = \langle c, \langle x_0, \ldots, x_{n-1} \rangle \rangle \in \mathfrak{T}(\dot{\vec{A}})$ in a structure $M = \langle X, \vec{A} \rangle$ as follows:

Let a be an assignment in X whose domain contains the free variables of t. Define an extension $\tilde{a}$ of a by setting:

$$\tilde{a}(x) = \begin{cases} a(x) & \text{if } x \neq \Phi, x \in \mathrm{dom}(a), \\ X & \text{if } x = \Phi. \end{cases}$$

Then I set:

$$\mathtt{val}^M[t](a) := (\mathtt{val}^{\vec{A}}[c])(\tilde{a}(x_0), \ldots, \tilde{a}(x_{n-1})).$$

If $\tilde{M} = \langle M, \vec{B} \rangle$ is a structure enhanced by additional predicates, I set:

$$\mathtt{val}^{\tilde{M}}[t](a) = \mathtt{val}^M[t](a).$$

Lemma 3.3. *Fix two lists of predicate symbols, $\dot{\vec{A}}$ and $\dot{\vec{B}}$. Then there is a recursive function $T = T_{\dot{\vec{A}};\dot{\vec{B}}}$ with the following property:*

Let $\vec{A}$ and $\vec{B}$ be interpretations of $\dot{\vec{A}}$ and $\dot{\vec{B}}$. Let X be a transitive set closed under functions rudimentary in $\vec{A}$, and let $\vec{A}, \vec{B} \subseteq X$. Set $X' = \mathrm{rud}_{\vec{A}}(X)$, and define $M := \langle X, \vec{A}, \vec{B} \rangle$, $M' := \langle X', \vec{A}, \vec{B} \rangle$. Let ϕ be a Σ_0-formula with free variables $v_0, \ldots, v_{n-1}$. Let $a = \{i_0, \ldots, i_{m-1}\} \in [n]^m$. For each $j < m$, let $t_j \in \mathfrak{T}(\dot{\vec{A}})$, such that no free variable of t_j occurs as a bound variable in ϕ.

Then $\psi := T(\phi, v_{i_0}, t_0, \ldots, v_{i_{m-1}}, t_{m-1})$ is a Σ_ω-formula with the following property: If $\vec{w} = w_0, \ldots, w_{m'-1}$ is an enumeration of $\{v_k \mid k \in n \setminus a\}$, then the set of free variables of ψ is contained in $\{\vec{w}\} \cup \bigcup_{j<m} \mathrm{Fr}\,(t_j)$ (here, repetitions may occur). Further, for any assignment b of the free variables of ψ with values in X, we have:

$$M' \models \phi[b'] \iff M \models \psi[b],$$

where $b' = b[({}^{v_{i_0}}/_{\mathtt{val}^M[t_0](b)}), \ldots, ({}^{v_{i_{m-1}}}/_{\mathtt{val}^M[t_{m-1}](b)})]$. *Hence, one might very well write:*

$$\psi = \phi(({}^{v_{i_0}}/_{t_0}), \ldots, ({}^{v_{i_{m-1}}}/_{t_{m-1}})).$$

Proof. Let $\langle \phi, v_{i_0}, t_0, \ldots, v_{i_{m-1}}, t_{m-1}\rangle$ be a suitable argument as in the lemma. For $l < m$, let $t_l = \langle c_l, \langle x_0^l, \ldots, x_{n_l-1}^l\rangle\rangle$.

The argument consists of three steps: First, ϕ will be transformed in such a way that it doesn't contain a subformula of the form $\dot{A}_k(v_i)$ $(1 \le k \le p)$ or $\dot{B}_l(v_i)$ $(1 \le l \le q)$ (for any $i \in a$). Then this formula is transformed using the substitution function Sub from Lemma 2.4. Finally, this formula is transformed into a Σ_ω formula that "expresses over M what the formula we started with expressed over M'".

The first transformation, T_1, is defined by recursion, for arbitrary Σ_0 formulae ψ. Given ψ, fix z to be the least variable that doesn't occur in ψ. Now define, by recursion on (not necessarily proper) subformulae of ψ, a function T_1^ψ.

For $\bar{\psi} \equiv C(v)$ $(C \in \{\dot{\vec{A}}, \dot{\vec{B}}\})$, where $v \in \{v_i \mid i \in a\}$, let

$$T_1^\psi(\bar{\psi}) \equiv \exists w \in z \quad w = v \ \wedge \ C(w).$$

The other cases of atomic formulae are trivial, i.e., these formulae remain unchanged by T_1^ψ. The expansion of T_1^ψ to Boolean combinations and to bounded quantifications is as usual. So the only actual change occurs in this one atomic case. Finally, let $T_1(\psi) = T_1^\psi(\psi)$.

The idea is that in the end, the value X will be substituted for z. Note that for an assignment h of the free variables of ψ with values in X, it is the case that:

$$(M' \models \psi[h]) \iff (\langle \mathrm{V}, \vec{A}, \vec{B}\rangle \models T_1(\psi)[h({}^z/_X)]).$$

Now let's return to the specific Σ_0 formula ϕ from the beginning of the proof. Let $\phi' = T_1(\phi)$. Choose (effectively) new variables $\vec{y}^0, \ldots, \vec{y}^{m-1}$, where $\vec{y}^j = y_0^j, \ldots, y_{n_j-1}^j$ (so $\vec{y}^j$ has the same length as $\vec{x}^j$), and form:

$$\tilde{\phi} := \mathrm{Sub}(\phi', v_{i_0}, c_0, \vec{y}^0, \ldots, v_{i_{m-1}}, c_{m-1}, \vec{y}^{m-1}).$$

Call the variables z and y_k^j with $x_k^j = \Phi$ *temporary.* Let $\vec{w}$, b and b' be as in the statement of the lemma, and define an assignment $\tilde{b}$ of the free variables of $\tilde{\phi}$ other than z by setting

$$\tilde{b}(v) = \begin{cases} X & \text{if } v \text{ is temporary and different from } z, \\ b(x_l^k) & \text{if } v = y_l^k \text{ and } x_l^k \neq \Omega, \\ b(v) & \text{otherwise.} \end{cases}$$

Then $\mathtt{val}^M[t_i](b) = \mathtt{val}[c_i](\tilde{b}(\vec{y}^i))$ for $i < m$, and so,

$$b' = b[({}^{v_{i_0}}/_{\mathtt{val}[c_0](\tilde{b}(\vec{y}^0))}) \cdots ({}^{v_{i_{m-1}}}/_{\mathtt{val}[c_{i_{m-1}}](\tilde{b}(\vec{y}^{m-1}))})].$$

In particular, b' and $\tilde{b}[({}^{v_{i_0}}/_{\mathtt{val}[c_0](\tilde{b}(\vec{y}^0))}) \cdots ({}^{v_{i_{m-1}}}/_{\mathtt{val}[c_{i_{m-1}}](\tilde{b}(\vec{y}^{i_{m-1}}))})]$ agree on the free variables of ϕ' (other than z, which is not in the domain of any of these assignments). So it's clear by Lemma 2.4 that

$$\langle \mathrm{V}, \vec{A}, \vec{B} \rangle \models \phi'[b'({}^z/_X)] \iff \langle \mathrm{V}, \vec{A}, \vec{B} \rangle \models \tilde{\phi}[\tilde{b}({}^z/_X)].$$

Note that these two formulae are Σ_0, so that they are satisfied in the universe iff they are satisfied in M'.

In order to pull $\tilde{\phi}$ down to M, I have to transform it into a Σ_ω formula in which the temporary variables don't occur freely anymore, and which is satisfied in M iff $\tilde{\phi}$ is satisfied in the real world, when all temporary variables are assigned the value X and the variables occurring in both formulae are assigned the same values. I define this transformation, $T_2(\psi)$, again by recursion on ψ.

Case 1: $\psi \equiv \dot{A}_k(x)$ $(1 \le k \le p)$ or $\psi \equiv \dot{B}_l(x)$ $(1 \le l \le q)$.

If x is not temporary, then ψ remains unchanged. Otherwise, I set: $T_2(\psi) \equiv \forall x \quad x \neq x$. This definition works, since I demanded that $\vec{A}$ and $\vec{B}$ are subsets of X.

Case 2: $\psi \equiv x \in y$.

Case 2.1: y is temporary, while x is not.

Then let $T_2(\psi) \equiv (x = x)$.

Case 2.2: x is temporary.

Then let $T_2(\psi) \equiv (\forall x \quad x \neq x)$.

Case 2.3: Neither x nor y are temporary.

Then ψ remains unchanged.

Case 3: $\psi \equiv x = y$.

Case 3.1: Exactly one of the variables x and y is temporary.

Let $v \in \{x, y\}$ be the variable which is not temporary. Then $T_2(\psi) = (v \neq v)$.

Case 3.2: Both x and y are temporary.

Then let $T_2(\psi) \equiv \forall x \quad x = x$.

Case 3.3: Neither x nor y is temporary.

Then ψ remains unchanged.

This defines T_2 for atomic formulae. The definition for Boolean combinations of formulae for which T_2 is already defined is as usual. The only remaining case of interest is

Case 4: $\psi \equiv \exists x \in y \quad \bar{\psi}$, where $T_2(\bar{\psi})$ is already defined.

If y isn't temporary, then let $T_2(\psi) \equiv \exists x \in y \quad T_2(\bar{\psi})$. In the other case I set:

$$T_2(\psi) \equiv \exists x \quad T_2(\bar{\psi}).$$

Note that in the formula I'm interested in, namely in $\tilde{\phi}$, only free variables can be temporary, so that it is irrelevant whether or not x is temporary in the current case.

The case of universal quantification is reduced in the usual way to existential quantification and negation. So this completes the definition of T_2.

So now I can define:

$$\psi \equiv T_2(\tilde{\phi})({}^{\langle y^j_k \mid j<m \wedge k<n_j \wedge x^j_k \neq \Phi\rangle}/_{\langle x^j_k \mid j<m \wedge k<n_j \wedge x^j_k \neq \Phi\rangle}).$$

So backtracking the definition, we have:

$$\begin{array}{rcl} \psi & \equiv & T(\phi, v_{i_0}, c_0, \langle \vec{x}^0 \rangle, \ldots, v_{i_{m-1}}, c_{m-1}, \langle \vec{x}^{m-1} \rangle) \\ & \equiv & T_2(\mathrm{Sub}(T_1(\phi), v_{i_0}, c_0, \vec{y}^0, \ldots, v_{i_{m-1}}, c_{m-1}, \vec{y}^{m-1})) \\ & & \quad ({}^{\langle y^j_k \mid j<m \wedge k<n_j \wedge x^j_k \neq \Phi\rangle}/_{\langle x^j_k \mid j<m \wedge k<n_j \wedge x^j_k \neq \Phi\rangle}), \end{array}$$

which obviously is recursive.

Let's check that ψ has the desired properties. Let b, b' and $\tilde{b}$ be as before.

Then it follows from the properties of T_2 that:

$$
\begin{aligned}
M \models \psi[b] \quad &\iff \quad M \models (T_2(\tilde{\phi})({}^{\langle y^j_k \mid j<m \wedge k<n_j \wedge x^j_k \neq \Phi\rangle}/_{\langle x^j_k \mid j<m \wedge k<n_j \wedge x^j_k \neq \Phi\rangle}))[b] \\
&\iff \quad M \models T_2(\tilde{\phi})[b({}^{\langle y^j_k \mid j<m \wedge k<n_j \wedge x^j_k \neq \Phi\rangle}/_{\langle b(x^j_k) \mid j<m \wedge k<n_j \wedge x^j_k \neq \Phi\rangle})] \\
&\iff \quad \langle \mathrm{V}, \vec{A}\rangle \models \tilde{\phi}[\tilde{b}({}^z/_X)] \\
&\iff \quad \langle \mathrm{V}, \vec{A}\rangle \models \phi'[b'({}^z/_X)] \\
&\iff \quad \langle \mathrm{V}, \vec{A}\rangle \models \phi[b'] \\
&\iff \quad M' \models \phi[b'].
\end{aligned}
$$

This completes the proof. □

4 Extender Ultrapowers of Successor Structures

In this section, I would like to give an application of the machinery developed thus far to extender ultrapowers of a structure in comparison to extender ultrapowers of its rud closure. In order to state the result precisely, I need some definitions. I shall adopt Jensen's view of extenders, see [Jen97].

Definition 4.1. Let $M = \langle X, \vec{A}, \vec{B}\rangle$ be a transitive structure which is closed under functions which are rudimentary in $\vec{A}$, and assume that $\vec{A}, \vec{B} \subseteq M$. Let $\kappa \in M$ be an ordinal which is primitive recursively closed, and assume that whenever $X_1, \ldots, X_n$ are in $\mathcal{P}(\kappa) \cap M$, $\nu_1, \ldots, \nu_n < \kappa$ and C is primitive recursive in the predicates $\vec{X}$ and the parameters $\vec{\nu}$, then $C \cap \kappa \in \mathcal{P}(\kappa) \cap M$. Then F is a (κ, λ)-extender on M if $\kappa < \lambda$, λ is primitive recursively closed, F is a function with domain $\mathcal{P}(\kappa) \cap M$ and $\mathrm{ran}(F) \subseteq \mathcal{P}(\lambda)$, such that whenever $X_1, \ldots, X_n \in \mathcal{P}(\kappa) \cap M$ and $C_{\vec{X}} \subseteq \mathrm{On}$ is primitive recursive in $\vec{X}$, then $F(C_{\vec{X}} \cap \kappa) = C_{F(\vec{X})} \cap \lambda$. κ is the *critical point* of F and λ is the *length* of F.

The following definition is designed to capture the structures to which the main theorem of this section applies. These structures make it possible to form fully elementary external extender ultrapowers.

Definition 4.2. A structure $M = \langle X, \vec{A}, \vec{B}\rangle$ is *definably well-ordered* iff it has a well-order which is definable in the structure. This means that there is a formula $\phi(x, y, z)$ in the language of set theory with additional unary predicate symbols $\dot{\vec{A}}, \dot{\vec{B}}$ such that there is an element $p \in X$, so that the set

$$\{\langle a, b\rangle \mid M \models \phi(a, b, p)\}$$

is a well-order of M.

I'll now describe what I call the (external) Σ_ω extender ultrapower.

Definition 4.3. Let $M = \langle X, \vec{A}, \vec{B}\rangle$ be $\mathrm{rud}_{\vec{A}}$-closed with $\vec{A}, \vec{B} \subseteq M$. Let F be a (κ, λ)-extender on M. Assume that $\mathcal{P}(\kappa) \cap \underset{\sim}{\Sigma}_\omega(M) = \mathcal{P}(\kappa) \cap M$. Then

$$\pi : M \longrightarrow_F^{\Sigma_\omega} N$$

expresses the following statements:

1. $\pi : M \longrightarrow_{\Sigma_\omega} N$,
2. N is transitive,
3. for $x \in \mathcal{P}(\kappa) \cap M$, $\pi(x) \cap \lambda = F(x)$,
4. for any function $f : \kappa \longrightarrow M$ which is definable over M in parameters $\vec{a}$, let $\pi(f)$ be the function defined over N by the same formula, in the parameters $\pi(\vec{a})$. Then
$$|N| = \{\pi(f)(\alpha) \mid f \in ({}^\kappa M) \cap \underset{\sim}{\Sigma}_\omega(M) \wedge \alpha < \lambda\}.$$

Analogously, if E is an extender on $\bar{M}$, then I write $\sigma : \bar{M} \longrightarrow_E \bar{N}'$ to express that $\bar{N}'$ is the usual extender ultrapower of $\bar{M}$ by E, formed with functions which are elements of $\bar{M}$, and that σ is the canonical embedding. This kind of ultrapower is sometimes referred to as a Σ_0-ultrapower.

The construction of Σ_ω-ultrapowers is an adaptation of the construction of fine structural ultrapowers.

Definition 4.4. Let $M = \langle X, \vec{A}, \vec{B}\rangle$ be $\mathrm{rud}_{\vec{A}}$-closed with $\vec{A}, \vec{B} \subseteq M$. Let F be a (κ, λ)-extender on M. Let

$$\begin{aligned} \Gamma^\omega(M,\kappa) &:= \{h \in \underset{\sim}{\Sigma}_\omega(M) \mid \exists n < \omega \quad h : \kappa^n \longrightarrow |M|\}, \\ D^\omega(M,\kappa,\lambda) &:= \{\langle \vec{\alpha}, f\rangle \mid f \in \Gamma^\omega(M,\kappa) \ \wedge \ f : \kappa^{\mathrm{lh}(\vec{\alpha})} \longrightarrow |M| \ \wedge \ \vec{\alpha} < \lambda\}. \end{aligned}$$

Define a relation $\sim$ on $D^\omega = D^\omega(M,\kappa,\lambda)$ by

$$\langle \vec{\alpha}, f\rangle \sim \langle \vec{\beta}, g\rangle \iff \prec\vec{\alpha}, \vec{\beta}\succ \in F(\{\prec\vec{\gamma}, \vec{\delta}\succ \mid \vec{\gamma}, \vec{\delta} < \kappa \ \wedge \ f(\vec{\gamma}) = g(\vec{\delta})\}).$$

Define also a "pseudo $\in$-relation" on D^ω:

$$\langle \vec{\alpha}, f\rangle E \langle \vec{\beta}, g\rangle \iff \prec\vec{\alpha}, \vec{\beta}\succ \in F(\{\prec\vec{\gamma}, \vec{\delta}\succ \mid \vec{\gamma}, \vec{\delta} < \kappa \ \wedge \ f(\vec{\gamma}) \in g(\vec{\delta})\}).$$

Note that these definitions can be made, as F measures all $\bar{M}$-definable subsets of κ.

The following is standard:

Lemma 4.5. *In the notation of the previous definition, $\sim$ is a congruence relation with respect to E: If $\langle\vec{\alpha},f\rangle \sim \langle\vec{\alpha}',f'\rangle$ and $\langle\vec{\beta},g\rangle \sim \langle\vec{\beta}',g'\rangle$, then*

$$\langle\vec{\alpha},f\rangle E\langle\vec{\beta},g\rangle \iff \langle\vec{\alpha}',f'\rangle E\langle\vec{\beta}',g'\rangle.$$

It is also a congruence relation with respect to the other predicates A'_i, B'_i, in the same sense:

$$A'_i(\langle\vec{\alpha},f\rangle) \iff A'_i(\langle\vec{\alpha}',f'\rangle) \quad \textit{and} \quad B'_j(\langle\vec{\alpha},f\rangle) \iff B'_j(\langle\vec{\alpha}',f'\rangle)$$

So the following definition is correct:

Definition 4.6. $\mathbb{D}^\omega(M,F) = \langle D^\omega(M,\kappa,\lambda)/{\sim}, E/{\sim}, \vec{A}'/{\sim}, \vec{B}'/{\sim}\rangle$, the set of $\sim$-equivalence classes of $D^\omega(\bar{M},\kappa,\lambda)$, where it is stipulated that

$$\begin{aligned} [\vec{\alpha},f]E/{\sim}[\vec{\beta},g] &\iff \langle\vec{\alpha},f\rangle E\langle\vec{\beta},g\rangle, \\ A'_i/{\sim}([\vec{\alpha},f]) &\iff A'_i(\langle\vec{\alpha},f\rangle), \\ B'_j/{\sim}([\vec{\alpha},f]) &\iff B'_j(\langle\vec{\alpha},f\rangle), \end{aligned}$$

where $[\vec{\alpha},f]$ denotes the $\sim$-equivalence class of $\langle\vec{\alpha},f\rangle \in \Gamma^\omega(M,\kappa,\lambda)$.

There is a version of Łoś's theorem for Σ_ω-ultrapowers of definably well-ordered structures:

Theorem 4.7. *Suppose that M is as above, with the additional assumption that M is definably well-ordered. Let $\mathbb{D}^\omega = \mathbb{D}^\omega(M,F)$. Let $\phi(x_0,\ldots,x_{n-1})$ be a formula, and let $[\vec{\alpha}^0,f^0],\ldots,[\vec{\alpha}^{n-1},f^{n-1}]$ be elements of $\mathbb{D}^\omega$. Then the following equivalence holds:*

$$\mathbb{D}^\omega \models \phi([\vec{\alpha}^0,f^0],\ldots,[\vec{\alpha}^{n-1},f^{n-1}])$$

$$\iff$$

$$\prec\vec{\alpha}^0,\ldots,\vec{\alpha}^{n-1}\succ \in F(\{\prec\vec{\beta}^0,\ldots,\vec{\beta}^{n-1}\succ \mid M \models \phi(f^0(\vec{\beta}^0),\ldots,f^{n-1}(\vec{\beta}^{n-1}))\}),$$

where $\dot{\in}$ is interpreted as $E/{\sim}$ in $\mathbb{D}^\omega$, and $\dot{A}_i$, $\dot{B}_j$ are interpreted as $A'_i/{\sim}$ and $B'_j/{\sim}$, respectively.

Proof. By induction on formulae. The point is that definable Skolem functions are available, as $\bar{M}$ is definably well-ordered. □

Theorem 4.8. *In the above notation, assume that $\mathbb{D}^\omega(M,F)$ is well-founded, and let N be its transitive isomorph. Define $\pi : |M| \longrightarrow |N|$ by setting $\pi(x) := [0,\mathrm{const}_x]$. If M is definably well-ordered, then*

$$\pi : M \longrightarrow_F^{\Sigma_\omega} N.$$

I also write $\mathrm{Ult}^{\Sigma_\omega}(M,F)$ for N.

Proof. This follows immediately from Łoś's theorem. □

Now I am ready to state the main result of this section.

Theorem 4.9. *Let $\dot{\vec{A}}, \dot{\vec{B}}$ be predicate symbols with interpretations $\vec{A}, \vec{B}$. Let $\bar{X}$ be a transitive set which is closed under functions rudimentary in $\vec{A}$, such that $\vec{A}, \vec{B} \subseteq \bar{X}$. Let $\bar{M} = \langle \bar{X}, \vec{A}, \vec{B} \rangle$ be definably well-ordered. Let $X := \mathrm{rud}_{\vec{A}}(\bar{X})$ and $M = \langle X, \vec{A}, \vec{B} \rangle$, and let $\underset{\sim}{\Sigma}_\omega(\bar{M}) = X \cap \mathcal{P}(\bar{X})$.*[3] *Let F be an extender on $\bar{M}$ and M.*[4] *Let*

$$\bar{\pi} : \bar{M} \longrightarrow_F^{\Sigma_\omega} \bar{M}',$$

where $\bar{M}' = \langle \bar{X}', \vec{A}', \vec{B}' \rangle$ is transitive. Then the following is a correct definition of a function π:

$$\pi(\mathtt{val}^{\bar{M}}[t](\vec{a})) := \mathtt{val}^{\bar{M}'}[t](\bar{\pi}(\vec{a})),$$

where $t \in \mathfrak{T}(\dot{\vec{A}})$ and $\vec{a} \in \bar{X}$ is an assignment of its free variables.

Set $X' := \mathrm{rud}_{\vec{A}'}(\bar{X}')$ and $M' := \langle X', \vec{A}', \vec{B}' \rangle$. Then

$$\pi : M \longrightarrow_F M' \quad \textit{and} \quad \bar{\pi} \subseteq \pi.$$

Proof. The proof will show that the definition of π makes sense even if $\bar{\pi}$ is not an extender embedding but an arbitrary elementary embedding. π will then be Σ_0 preserving and cofinal, hence Σ_1 preserving (even Q-preserving; cf. [Zem02, p. 3]). It is also known (and implicit in the following proof) that π is the only Σ_0 preserving embedding from M into M' extending $\bar{\pi}$; cf. [Dev84, Lemma 1.19]. What's new here is that π is again an extender ultrapower.

First, note that $\bar{X}'$ is closed under functions rudimentary in $\vec{A}'$. This can be seen as follows: If c is a code for a function rudimentary in $\dot{\vec{A}}$, then we have:

$$\begin{array}{rl} & \forall \vec{a} \in \bar{M} \exists b \in \bar{M} \quad b = \mathtt{val}^{\vec{A}}[c](\vec{a}) \\ \iff & \bar{M} \models \forall \vec{x} \exists y \quad \mathrm{Sub}(y = v, v, c, \langle \vec{x} \rangle) \\ \iff & \bar{M}' \models \forall \vec{x} \exists y \quad \mathrm{Sub}(y = v, v, c, \langle \vec{x} \rangle) \\ \iff & \forall \vec{a} \in \bar{M}' \exists b \in \bar{M}' \quad b = \mathtt{val}^{\vec{A}'}[c](\vec{a}). \end{array}$$

Now I will verify the correctness of the definition of π. Assume we have two representations of some $x \in X$:

$$x = \mathtt{val}^{\bar{M}}[t_1](\vec{a}^1) = \mathtt{val}^{\bar{M}}[t_2](\vec{a}^2).$$

[3]If $\vec{B}$ is empty, then it is a general fact that $\underset{\sim}{\Sigma}_\omega(\bar{M}) = X \cap \mathcal{P}(\bar{X})$, see [Jen72, Cor. 1.7]. But otherwise, this need not be true, since X is the rudimentary closure of $\bar{X} \cup \{\bar{X}\}$ only under functions which are rudimentary in $\vec{A}$. So $\underset{\sim}{\Sigma}_\omega(\bar{M})$ will contain each B_i as an element, while this is not necessarily true of X.

[4]Note that, letting $\kappa = \mathrm{crit}(F)$, this implies that $\mathcal{P}(\kappa) \cap \bar{X} = \mathrm{dom}(F) = \mathcal{P}(\kappa) \cap X$.

Let $\vec{x}^1 = \mathrm{Fr}(t_1)$ and $\vec{x}^2 = \mathrm{Fr}(t_2)$. By renaming the free variables, we may assume that $\mathrm{Fr}(t_1) \cap \mathrm{Fr}(t_2) = \emptyset$. Then we have:

$$\begin{array}{rl} & M \models (v = w)[({}^{v}/_{\mathtt{val}^{\bar{M}}[t_1](\vec{a}^1)}), ({}^{w}/_{\mathtt{val}^{\bar{M}}[t_2](\vec{a}^2)})] \\ \iff & \bar{M} \models \underbrace{T_{\vec{A};\vec{B}}(v = w, v, t_1, w, t_2)[({}^{\vec{x}^1}/_{\vec{a}^1}), ({}^{\vec{x}^2}/_{\vec{a}^2})]}_{\psi[\vec{a}^1, \vec{a}^2]} \\ \iff & \bar{M}' \models \psi[\bar{\pi}(\vec{a}^1), \bar{\pi}(\vec{a}^2)] \\ \iff & M' \models (v = w)[({}^{v}/_{\mathtt{val}^{\bar{M}'}[t_1](\bar{\pi}(\vec{a}^1))}), ({}^{w}/_{\mathtt{val}^{\bar{M}'}[t_2](\bar{\pi}(\vec{a}^2))})], \end{array}$$

so $\mathtt{val}^{\bar{M}'}[t_1](\bar{\pi}(\vec{a}^1)) = \mathtt{val}^{\bar{M}'}[c_2](\bar{\pi}(\vec{a}^2))$. Hence, the definition of π is independent of the representation of x.

Now it is obvious that $\bar{\pi} \subseteq \pi$, since for $a \in \bar{X}$, we have

$$\pi(a) = \pi(\mathtt{val}^{\bar{M}}[\pi_0^1](a)) = \mathtt{val}^{\bar{M}'}[\pi_0^1](\bar{\pi}(a)) = \bar{\pi}(a);$$

as a reminder: π_0^1 is the code for the projection of one-tuples onto the first coordinate, i.e., the identity.

I want to show that $\pi : M \longrightarrow_F M'$.

The proof that showed that π is correctly defined also shows that π is Σ_0 preserving; instead of "$(v = w)$" one can use an arbitrary Σ_0 formula. Also, let $\kappa = \mathrm{crit}(F)$ and $\lambda = \mathrm{lh}(F)$. Then, since $\bar{\pi} \subseteq \pi$ and $\bar{\pi} : \bar{M} \longrightarrow_F^{\Sigma_\omega} \bar{M}'$, it follows that for $x \in \mathcal{P}(\kappa) \cap M$, $\pi(x) \cap \lambda = \bar{\pi}(x) \cap \lambda = F(x)$. Set (in analogy to Definition 4.4):

$$\begin{array}{rcl} \Gamma^0(M, \kappa) & := & \{h \in M \mid \exists n < \omega \quad h : \kappa^n \longrightarrow M\}, \\ D^0(M, \kappa, \lambda) & := & \{\langle \vec{\alpha}, f \rangle \mid f \in \Gamma^0(M, \kappa) \wedge \mathrm{dom}(f) = \kappa^{\mathrm{lh}(\vec{\alpha})} \wedge \vec{\alpha} < \lambda\}. \end{array}$$

I have to show that

($\star$) $X' = \{\pi(f)(\vec{\alpha}) | \langle \vec{\alpha}, f \rangle \in D^0(M, \kappa, \mathrm{lh}(F))\}$.

First, for $f \in \Gamma^\omega(\bar{M}, \kappa)$, I am going to define the value $\bar{\pi}(f)$, as in Definition 4.3: If f is defined in $\bar{M}$ by $\phi(y, x, z)$, i.e., if there is a $p \in \bar{M}$ such that for all $b, a \in |\bar{M}|$,

$$b = f(a) \iff \bar{M} \models \phi[b, a, p],$$

then let $\bar{\pi}(f)$ be the function defined in $\bar{M}'$ by ϕ in the parameter $\bar{\pi}(p)$. That this is a correct definition of a function is a consequence of the fact that $\bar{\pi}$ is elementary. We get:

(1) *For* $f \in \Gamma^\omega(\bar{M}, \kappa)$, $\bar{\pi}(f) = \pi(f)$.

Proof of (1). Let $b = f(a) \iff \bar{M} \models \phi[b, a, p]$. Since $f \in \mathbf{\Sigma}_\omega(\bar{M})$, it follows that $f \in X$, by assumption. So let t be a rudimentary term, and let $\vec{a} \in \bar{X}$ be such that

$$f = \mathtt{val}^{\bar{M}}[t](\vec{a}).$$

Let $d = \mathrm{dom}(f) \in \bar{X}$. So $d = \kappa^n$ for some $n < \omega$.

Then $\mathrm{dom}(\bar{\pi}(f)) = \bar{\pi}(d) = \pi(d)$. For arbitrary $b, b' \in \bar{X}$, we get:

$$\begin{aligned} \bar{M} \models \phi[b', b, p] &\iff b' = f(b) \\ &\iff M \models \langle b', b\rangle \in \mathtt{val}^M[t](\vec{a}) \\ &\iff \bar{M} \models \psi[b', b, \vec{a}]. \end{aligned}$$

Here,

$$\psi = T_{\vec{A};\vec{B}}(\langle u, v\rangle \in w, w, t).$$

So we have:

$$\begin{aligned} &\bar{M} \models \forall b', b \quad (\psi[b', b, \vec{a}] \longleftrightarrow \phi[b', b, p]) \\ \iff &\bar{M}' \models \forall b', b \quad (\psi[b', b, \bar{\pi}(\vec{a})] \longleftrightarrow \phi[b', b, \bar{\pi}(p)]). \end{aligned}$$

Here I used the full elementarity of $\bar{\pi}$. Hence, for arbitrary $b', b \in \bar{X}'$, the following holds:

$$\begin{aligned} \pi(f)(b) = b' &\iff \bar{M}' \models \psi[b', b, \bar{\pi}(\vec{a})] \\ &\iff \bar{M}' \models \phi[b', b, \bar{\pi}(p)] \\ &\iff \bar{\pi}(f)(b) = b'. \end{aligned}$$

This is what was claimed. $\square_{(1)}$

The proof of (1) actually shows that if for $X \in \underset{\sim}{\Sigma}_\omega(\bar{M})$ we define $\bar{\pi}(X)$ to be the set defined over $\bar{M}'$ by the same formula that defined X over $\bar{M}$, with the parameters moved by $\bar{\pi}$, then $\bar{\pi}(X) = \pi(X)$.

Now I can approach the proof of ($\star$). To this end, let $x \in X'$. Then

$$x = \mathtt{val}^{\bar{M}'}[t](\vec{b}),$$

for some term $t \in \mathfrak{T}(\vec{A})$ and elements $\vec{b} = b_1, \ldots, b_n \in \bar{X}'$. I may assume that $t = \langle c, \langle \vec{x}, \Omega\rangle\rangle$, where $\vec{x}$ are the free variables of t and $c \in \mathfrak{C}(\vec{A})$.

Since $\bar{\pi} : \bar{M} \longrightarrow_F^{\Sigma_\omega} \bar{M}'$, there are $\langle \vec{\alpha}^1, f_1\rangle, \ldots, \langle \vec{\alpha}^n, f_n\rangle \in D^\omega(\bar{M}, \kappa, \mathrm{lh}(F))$ such that we have:

$$b_i = \bar{\pi}(f_i)(\vec{\alpha}^i)$$

for $1 \le i \le n$. So by (1), $b_i = \pi(f_i)(\vec{\alpha}^i)$.

Let $f_i : \kappa^{l_i} \longrightarrow \bar{X}$. Set: $m = \sum_{1 \le i \le n} l_i$. Now define a function $g : \kappa^m \longrightarrow X$ by:

$$g(\vec{\gamma}^1, \ldots, \vec{\gamma}^n) = \mathtt{val}^{\bar{M}}[t](f_1(\vec{\gamma}^1), \ldots, f_n(\vec{\gamma}^n)).$$

I want to show that $x = \pi(g)(\vec{\alpha}^1, \ldots, \vec{\alpha}^n)$. A first step in this direction is:

(2) $g \in X$.

Proof of (2). The function

$$h_1(\vec{z}^1, \ldots, \vec{z}^n, g_1, \ldots, g_n, v) = \langle g_1(\langle \vec{z}^1 \rangle), \ldots, g_n(\langle \vec{z}^n \rangle), v \rangle$$

is (uniformly) rudimentary by [Jen72, 1.1.(d),1.3.(b)].

Let $h = \mathtt{val}^{\bar{M}}[t]$, that is, h is the function mapping $\vec{z}$ to $\mathtt{val}^{\bar{M}}[t](\vec{z})$. Then there is a function h' which is rudimentary in $\vec{A}$, so that for all $\vec{z}$, $h(\vec{z}) = h'(\vec{z}, \bar{X})$. Now

$$\begin{aligned} g &= \langle (h' \circ h_1)(\vec{\gamma}^1, \ldots, \vec{\gamma}^n, f_1, \ldots, f_n, \bar{X}) \mid \vec{\gamma}^1, \ldots, \vec{\gamma}^n \in \kappa^m \rangle \\ &= h_2(\kappa^m, \vec{f}, \bar{X}), \end{aligned}$$

where h_2 is rudimentary in $\vec{A}$. This shows that $g \in X$, as X is closed under functions rudimentary in $\vec{A}$, and $\vec{f}, \bar{X} \in X$. $\Box_{(2)}$

(3) $\pi(g)(\vec{\alpha}^1, \ldots, \vec{\alpha}^n) = x$.

Proof of (3). By (2), I can choose $\tilde{c} \in \mathfrak{C}(\dot{\vec{A}})$ and $\vec{d} \in \bar{X}$ so that

$$g = \mathtt{val}^{\vec{A}}[\tilde{c}](\vec{d}, \bar{X}).$$

Since $f_i \in X$, for $1 \le i \le n$, I can choose $c_i \in \mathfrak{C}(\dot{\vec{A}})$ and $\vec{d^i} \in \bar{X}$ in such a way that

$$f_i = \mathtt{val}^{\vec{A}}[c_i](\vec{d^i}, \bar{X}).$$

Let $k_i + 1$ be the arity of the term associated to c_i, i.e., let $\vec{d^i}$ be a k_i-tuple. Set:

$$p = \sum_{1 \le i \le n} k_i \text{ and } q = m + p + 1.$$

Using [Jen72, Prop. 1.3. (b), 1.1.(d)] it is easy to see that for $k < \omega$, the following function is rudimentary:

$$\mathtt{app}^k(f, x_0, \ldots, x_{k-1}) = \begin{cases} f(x_0, \ldots, x_{k-1}) & \text{if it exists,} \\ \emptyset & \text{otherwise.} \end{cases}$$

Let $\dot{\mathtt{app}}^k \in \mathfrak{C}$ be a code for $\mathtt{app}^k$, i.e., $\mathtt{app}^k = \mathtt{val}[\dot{\mathtt{app}}^k]$.

In the follwing, I will construct a code c' for an m-ary function rudimentary in $\dot{\vec{A}}$, so that "$\mathtt{val}^{\vec{A}}[c'] \upharpoonright \kappa^m = g$" – this has to be taken with a grain of salt. Formulating the exact relation forces us to go through the following notational hell:

$$
\begin{aligned}
&g(\vec{\gamma}^1,\ldots,\vec{\gamma}^n) = \\
&= \mathtt{val}^{\vec{A}}[c \circ (\dot{\mathtt{app}}^{l_1} \circ (c_1 \circ (\pi^q_m, \pi^q_{m+1}, \ldots, \pi^q_{k_1-1}, \pi^q_{q-1}), \pi^q_0, \ldots, \pi^q_{l_1-1}), \\
&\qquad\qquad \dot{\mathtt{app}}^{l_2} \circ (c_2 \circ (\pi^q_{m+k_1}, \pi^q_{m+k_1+1}, \ldots, \pi^q_{m+k_1+k_2-1}, \pi^q_{q-1}), \\
&\qquad\qquad\qquad\qquad \pi^q_{l_1}, \ldots, \pi^q_{l_1+l_2-1}), \\
&\qquad\qquad \ldots, \\
&\qquad\qquad \dot{\mathtt{app}}^{l_n} \circ (c_n \circ (\pi^q_{m+k_1+\ldots+k_{n-1}}, \pi^q_{m+k_1+\ldots+k_{n-1}+1}, \ldots, \\
&\qquad\qquad\qquad\qquad \pi^q_{m+k_1+\ldots+k_{n-1}+k_n-1}, \pi^q_{q-1}), \\
&\qquad\qquad\qquad\qquad \pi^q_{l_1+\ldots+l_{n-1}}, \pi^q_{l_1+\ldots+l_{n-1}+1}, \ldots, \pi^q_{l_1+\ldots+l_{n-1}+l_n-1}), \\
&\qquad\qquad \pi^q_{q-1})] \\
&\qquad\qquad (\vec{\gamma}^1,\ldots,\vec{\gamma}^n, \vec{d}^1, \ldots, \vec{d}^n, \bar{X}). \\
&:= \mathtt{val}^{\vec{A}}[c'](\vec{\gamma}^1,\ldots,\vec{\gamma}^n, \vec{d}^1, \ldots, \vec{d}^n, \bar{X}).
\end{aligned}
$$

For arbitrary $\vec{\gamma}^1,\ldots,\vec{\gamma}^n \in \kappa^m$, we get:

$$M \models (\mathtt{val}^{\vec{A}}[\tilde{c}](\vec{d}, \bar{X}))(\vec{\gamma}^1,\ldots,\vec{\gamma}^n) = \mathtt{val}^{\vec{A}}[c'](\vec{\gamma}^1,\ldots,\vec{\gamma}^n, \vec{d}^1, \ldots, \vec{d}^n, \bar{X}),$$

by definition of c'. This means,

$$\bar{M} \models \psi[\vec{\gamma}^1,\ldots,\vec{\gamma}^n, \vec{d}^1, \ldots, \vec{d}^n],$$

where

$$\psi(\vec{x},\vec{y}) = T_{\vec{A};\vec{B}}((v(\vec{x}) = w), v, \langle \tilde{c}, \langle \vec{y}, \Phi \rangle\rangle, w, \langle c', \langle \vec{x}, \vec{y}, \Phi\rangle\rangle).$$

So we get:

$$\bar{M} \models \forall \vec{\gamma}^1,\ldots,\vec{\gamma}^n \in \kappa \quad \psi[\vec{\gamma}^1,\ldots,\vec{\gamma}^n, \vec{d}^1, \ldots, \vec{d}^n],$$

and as $\bar{\pi}$ is elementary,

$$\bar{M}' \models \forall \vec{\gamma}^1,\ldots,\vec{\gamma}^n \in \bar{\pi}(\kappa) \quad \psi[\vec{\gamma}^1,\ldots,\vec{\gamma}^n, \bar{\pi}(\vec{d}^1), \ldots, \bar{\pi}(\vec{d}^n)],$$

in particular

$$\bar{M}' \models \psi[\vec{\alpha}^1,\ldots,\vec{\alpha}^n, \bar{\pi}(\vec{d}^1), \ldots, \bar{\pi}(\vec{d}^n)].$$

By definition of ψ, and applying Lemma 3.3, this means:

$$
\begin{aligned}
M' \models (\mathtt{val}^{\vec{A}'}[\tilde{c}](\bar{\pi}(\vec{d}), \bar{X}'))(\vec{\alpha}^1,\ldots,\vec{\alpha}^n) = \\
= \mathtt{val}^{\vec{A}'}[c'](\vec{\alpha}^1,\ldots,\vec{\alpha}^n, \bar{\pi}(\vec{d}^1), \ldots, \bar{\pi}(\vec{d}^n), \bar{X}').
\end{aligned}
$$

Unraveling the definition of c', we get:

$$\begin{aligned}
\pi(g)(\vec{\alpha}^1,\ldots,\vec{\alpha}^n) &= (\mathtt{val}^{\vec{A}'}[\tilde{c}](\bar{\pi}(\vec{d}),\bar{X}'))(\vec{\alpha}^1,\ldots,\vec{\alpha}^n)\\
&= \mathtt{val}^{\vec{A}'}[c'](\vec{\alpha}^1,\ldots,\vec{\alpha}^n,\bar{\pi}(\vec{d}^1),\ldots,\bar{\pi}(\vec{d}^n),\bar{X}')\\
&= \mathtt{val}^{\vec{A}'}[c]((\mathtt{val}^{\vec{A}'}[c_1](\bar{\pi}(\vec{d}^1),\bar{X}'))(\vec{\alpha}^1),\ldots\\
&\qquad \ldots,(\mathtt{val}^{\vec{A}'}[c_n](\bar{\pi}(\vec{d}^n),\bar{X}'))(\vec{\alpha}^n),\bar{X}')\\
&= \mathtt{val}^{\vec{A}'}[c](\pi(\mathtt{val}^{\vec{A}}[c_1](\vec{d}^1,\bar{X}))(\vec{\alpha}^1),\ldots,\\
&\qquad \ldots\pi(\mathtt{val}^{\vec{A}}[c_n](\vec{d}^n,\bar{X}))(\vec{\alpha}^n),\bar{X}')\\
&= \mathtt{val}^{\vec{A}'}[c](\pi(f_1)(\vec{\alpha}^1),\ldots,\pi(f_n)(\vec{\alpha}^n),\bar{X}')\\
&= \mathtt{val}^{\vec{A}'}[c](b_1,\ldots,b_n,\bar{X}')\\
&= x,
\end{aligned}$$

which was to be shown. This proves (3), ($\star$), and hence the theorem. □

In the following, I will provide the most familiar context in which the previous theorem can be applied. I will freely use fine structural concepts due to Jensen, for which [Zem02] serves as a basic reference.

Theorem 4.10. *Let M be an acceptable J-structure with $R^*_M \neq \emptyset$. Let F be an extender on M at (κ,λ), where $\rho^\omega_M > \kappa$, and let $\pi : M \longrightarrow^*_F N$. Then π is fully elementary.*

Proof. For $k < \omega$, let

$$\pi_k : M \longrightarrow^k_F N_k$$

be the k-ultrapower (see [Zem02, Section 3.5]). Let $p \in R^*_M$. Then $p \restriction k \in R^k_M$, and so, by [Zem02, Lemma 3.5.1], $\pi_k(p \restriction k) \in R^k_{N_k}$. Now define

$$\sigma_{m,n} : N_m \longrightarrow N_n$$

by setting $\sigma_{m,n}(\pi_m(f)(\vec{\alpha})) = \pi_n(f)(\vec{\alpha})$, for $m \leq n$, $f \in \Gamma^m(M,\kappa)$ (i.e., f a good $\Sigma^{(l)}_1(M)$-function, for some $l < m$), and $\vec{\alpha} < \lambda$. Then $\sigma_{m,n}$ is $\Sigma^{(m)}_0$-preserving, since the Łoś theorem holds for $\Sigma^{(m)}_0$-formulae. So if $\phi(x)$ is $\Sigma^{(m)}_0$, then we get:

$$\begin{aligned}
N_m \models \phi(\pi_m(f)(\vec{\alpha})) &\iff \vec{\alpha} \in F(\{\vec{\beta} < \kappa \mid M \models \phi(f(\vec{\beta}))\})\\
&\iff N_n \models \phi(\pi_n(f)(\vec{\alpha})).
\end{aligned}$$

Since $\sigma_{m,n}(\pi_m(p \restriction m)) = \pi_n(p \restriction m) \in R^m_{N_n}$, [Zem02, Lemma 1.5.2] shows that

$$\sigma_{m,n} : N_m \longrightarrow_{\Sigma_m} N_n,$$

and the same lemma yields that

$$\pi_m : M \longrightarrow_{\Sigma_m} N_m.$$

Now let $\pi : M \longrightarrow^*_F N$ be the $*$-ultrapower. Define

$$\sigma_n : N_n \longrightarrow N$$

by setting $\sigma_n(\pi_n(f)(\vec{\alpha})) = \pi(f)(\vec{\alpha})$. Then clearly,

$$\langle N, \langle \sigma_n \mid n < \omega \rangle \rangle = \text{dir lim}(\langle \langle N_m \mid m < \omega \rangle, \langle \sigma_{m,n} \mid m \leq n < \omega \rangle \rangle).$$

We have

$$\sigma_n \circ \pi_n = \pi,$$

for all $n < \omega$: $\sigma_n(\pi_n(x)) = \sigma_n(\pi_n(\text{const}_x)(0)) = \pi(\text{const}_x)(0) = \pi(x)$. Here is a commutative diagram clarifying the situation (for arbitrary $m < n < \omega$):

$$\begin{tikzcd}
& N_n \arrow[dr, "\sigma_n"] & \\
M \arrow[ur, "\pi_n"] \arrow[rr, "\pi"] \arrow[dr, "\pi_m"'] & & N \\
& N_m \arrow[uu, "\sigma_{m,n}"'] \arrow[ur, "\sigma_m"'] &
\end{tikzcd}$$

Obviously, since $\sigma_{n,l}$ is Σ_n-preserving, for all $l \geq n$, it follows that

$$\sigma_n : N_n \longrightarrow_{\Sigma_n} N.$$

So since also $\pi_n : M \longrightarrow_{\Sigma_n} N_n$, it follows that π is Σ_n-preserving, for every $n < \omega$, and hence that π is fully elementary. $\square$

Corollary 4.11. *Let $M = \mathrm{J}^{E^M}_{\alpha+1}$ be a premouse of height $\alpha + 1$. Let $\bar{M} = M||\alpha$, and let $G = E^M_\alpha$. Let F be an extender on $\bar{M}$, such that* $\text{crit}(F) < \rho^\omega_{\bar{M}}$. *Then F is an extender on M also. Let*

$$\bar{\pi} : \bar{M} \longrightarrow^*_F \bar{M}',$$

where $\bar{M}'$ is transitive. Let $\bar{M}' = \langle \mathrm{J}^{E'^{\bar{M}'}}_{\alpha'}, G' \rangle$, and let $M' = \langle \mathrm{J}^{E^{M'}}_{\alpha'+1}, \emptyset \rangle$, where E' is the extender sequence $E^{\bar{M}'} {}^\frown \langle G' \rangle$ (in the obvious sense). Then the following is a correct definition of a function $\pi : M \longrightarrow M'$:

$$\pi(\mathtt{val}^{\bar{M}}[t](\vec{a})) := \mathtt{val}^{\bar{M}'}[t](\bar{\pi}(\vec{a})),$$

where $t \in \mathfrak{T}(\dot{E}, \dot{G})$ *and* $\vec{a} \in |\bar{M}|$ *is an assignment of its free variables. Moreover,*

$$\pi : M \longrightarrow_F M' \quad \textit{and} \quad \bar{\pi} \subseteq \pi.$$

Proof. Since $\bar{M}$ is a proper initial segment of a premouse, it is sound, and so, in particular, it has a very good parameter. So by the previous theorem, $\bar{\pi} : \bar{M} \longrightarrow_{\Sigma_\omega} \bar{M}'$. But then it is clear that $\bar{\pi} : \bar{M} \longrightarrow_F^{\Sigma_\omega} \bar{M}'$ is the Σ_ω-ultrapower: Every member of $\bar{M}'$ is of the form $\bar{\pi}(f)(\vec{\xi})$, where f not only is definable in $\bar{M}'$, but is even a good $\Sigma_1^{(n)}$-function, for some $n < \omega$. Vice versa, if $f : \kappa^m \longrightarrow \bar{M}$ is a function which is definable over $\bar{M}$, then since $\bar{\pi}$ is fully elementary, it makes sense to let $\bar{\pi}(f) : \bar{\pi}(\kappa) \longrightarrow \bar{M}'$ be the function defined by the same formula over $\bar{M}'$, with the parameters used in the definition moved by $\bar{\pi}$. Then clearly $\bar{\pi}(f)(\vec{\xi}) \in \bar{M}'$, for all $\vec{\xi} < \lambda$.

It is easy to check that $|M| = \mathrm{rud}_{E^M \restriction \alpha, G}(|\bar{M}|)$. So, knowing that $\bar{\pi} : \bar{M} \longrightarrow_F^{\Sigma_\omega} \bar{M}'$ is the Σ_ω-ultrapower, Theorem 4.9 (with $\vec{A} = E^M \restriction \alpha, G$, and $\vec{B}$ empty) can be applied to get the desired conclusion. □

References

[Dev84] Keith J. Devlin. *Constructibility.* Springer, Berlin, 1984.

[Fuc] Gunter Fuchs. λ-structures and s-structures: Translating the models. In preparation.

[Fuc03] Gunter Fuchs. *λ-Strukturen und s-Strukturen.* PhD dissertation, Humboldt-University, Berlin, Germany, Department of Mathematics, May 2003. http://www.edoc.hu-berlin.de.

[Jen72] Ronald Jensen. The fine structure of the constructible hierarchy. *Annals of Mathematical Logic*, 4:229–308, 1972.

[Jen97] Ronald Jensen. A new fine structure for higher core models. Handwritten Notes, 1997.

[MS94] William J. Mitchell and John R. Steel. *Fine Structure and Iteration Trees.* Lecture Notes in Logic **3**. Springer, Berlin, 1994.

[Zem02] Martin Zeman. *Inner Models and Large Cardinals.* Springer, Berlin, 2002.