Theoretical Computer Science 110 (1993) 405–418
Elsevier

A semantic characterization of the well-typed formulae of λ-calculus

Thomas Forster

Department of Pure Mathematics and Mathematical Statistics, 16 Mill Lane, Cambridge CB2 1SB, UK

Communicated by M. Sintzoff
Received May 1990
Revised June 1991

Abstract

Forster, T., A semantic characterization of the well-typed formulae of λ-calculus, Theoretical Computer Science 110 (1993) 405–418.

A model-theoretic operation is characterized that preserves the property of being a model of typed λ-calculus (i.e., the result of applying it to a model of typed λ-calculus is another model of typed λ-calculus). An expression is *well-typed* iff the class of its models is closed under this operation.

1. Introduction

1.1. History

Type disciplines are meat and drink to computer scientists. People whose first encounter with them was as a means of avoiding Russell's paradox and its kin (which meant most people – or at least most logicians – in the days before theoretic computer science acquired its present importance in logic) are liable to think that they are ad hoc modifications which are justified by their usefulness for this purpose. This is a mistake. Most type distinctions in computer science arise in a quite different way. Consider (finite) product types, for example, which arise from ordered tuple functions. All we know about ordered pairs is that we have three functions: $\mathrm{pair}(x, y)$, $\mathrm{fst}(x)$ and $\mathrm{snd}(y)$ and the obvious algebraic theory for them. The typing discipline provided by product types very neatly characterises as ill-typed precisely those existential formulae that are not true in the free algebra. For example, the assertion '$(\exists x)(x = \mathrm{pair}(x, x))$' is

Correspondence to: T. Forster, Department of Pure Mathematics and Mathematical Statistics, 16 Mill Lane, Cambridge CB2 1SB, UK.

0304-3975/93/$06.00

ill-typed and false in the free algebra. Another way of putting this is to say that the ill-typed existential sentences are precisely those whose truth value is implementation-dependent. The apparent difference between these two ways of generating type distinctions (and there are others, for example one can always introduce a quotient type whenever there is an equivalence relation, as in the introduction of cardinal numbers) is that the type distinctions of set theory are invoked in order to prevent us slipping into paradox while the type distinctions given by product types are designed merely to prevent us asking silly questions to which the system cannot provide an answer (because it is implementation-dependent).

Despite these two apparently different geneses one can give a surprisingly unified treatment of these two syntactical devices, and it takes the form of a "preservation theorem" of the kind standard in model theory of the middle years of this century. One of the functions of model theory is to identify semantic properties ("What does the class of models of the formula look like?") with syntactic properties ("What does the formula look like") which tend to be easier to deal with. Preservation theorems are results of the form: "A formula Φ has syntactic property Γ iff the class of its models is closed under the operation Δ".

In [3] I gave a preservation theorem that characterizes the *stratified* formulae of set theory. I knew at that time that the result could be extended to characterize the well-typed formulae of λ-calculus (with function types but not product types) and announced this result as something which was to come, but I had not noticed that it could cover (finite) product types as well. The purpose of this article is to provide the correct generalisation of the machinery of Forster [3] to the polyadic case – something which it is easy to get wrong!

The preservation theorem of Forster [3] identifies stratified formulae of set theory as those formulae ϕ such that the class of all models of ϕ is closed under the construction of permutation models (in the sense of Rieger–Bernays, not Fränkel–Mostowski) for a particular kind of permutation, which I called *setlike*. The idea is very simple: If σ is a permutation of a set M (being the domain of a model $\mathcal{M} = \langle M, \in \rangle$) then we obtain a new structure $\mathcal{M}^\sigma$ with domain M and membership relation $x \in_\sigma y$ iff $x \in \sigma(y)$. This new structure is a *permutation model of* $\mathcal{M}$. The principal result of [3] is that a formula Φ of the language of set theory is equivalent to a stratified formula iff whenever $\mathcal{M}$ is a model of Φ (and σ is setlike) then $\mathcal{M}^\sigma$ is too.

Permutation models have a long history. If we have a model $\langle V, \in \rangle$ of ZF let σ be the transposition exchanging the empty set and its singleton. Then we define $x \in_\sigma y$ by $x \in \sigma`y$, and, in general, $\phi^\sigma$ as the result of replacing all atomic formulae $x \in y$ in $\phi$ by $x \in \sigma`y$. It turns out that in the model V^σ consisting of the old universe and the new membership relation $\in_\sigma$ the "old" empty set has become an object identical to its own singleton, and foundation has failed. As it happens all the other axioms of ZF are preserved. The new model is said to be a *permutation model* of the old.

This is in Rieger and Bernays. Scott [9] showed how any permutation model of a model of Quine's NF is also a model of NF, and since this was (and remains) the only useful way of transforming models of NF it was refined to an independence method by

Henson, Pétry and Hinnion.[1] Henson (working in NF) showed that permutation models preserve all stratified sentences. However, since in NF permutations of the universe can be *sets*, the universe itself is a set, and there is no bar on large classes being sets, Henson naturally considered permutations that are *sets*. As it happens there are some (unstratified) sentences that are preserved by permutations that are sets that are not preserved by all permutations – notably the whole of cardinal arithmetic. Permutations that are not sets of the model will preserve (in general) only those formulae that use two types only. Therefore, to obtain a version of Henson's theorem for which a proving a converse was feasible, we needed to find the right class of permutations: one that preserved neither too much nor too little. This class, the class of *setlike* permutations, was discovered by Forster [3]. The refinement of Henson's theorem we need is that stratified sentences are preserved by setlike permutations.

Now for the converse. The other missing concept that had to be found was that of a stratimorphism. A stratimorphism between two structures is a *family* of bijections between the two structures that in some sense preserves stratified information only. This was discovered independently by the author and Pétry in the 1970s. Although the name "stratimorphism" is the author's, it was Pétry who first saw the use to which these objects could be put. There is a celebrated lemma of Keisler – the Keisler ultrapower lemma – which is a very useful tool in model theory. The reader will recall that this lemma says that two structures are elementarily equivalent iff they have isomorphic ultrapowers. Pétry felt there should be a version of this theorem with "elementarily equivalent with respect to stratified formulae" instead of "elementarily equivalent" and "stratimorphic" instead of isomorphic, and he proved this in [7]. Pétry's stratified version of Keisler's ultrapower Lemma is the critical step in proving the converse to the refinement of Henson's Lemma, and the conjunction of these two is the main theorem of Forster [3].

The proof of that result depended on features of $\in$ which can be found elsewhere. In particular $\in$ is extensional: I can identify x if I know all the y such that $y \in x$. Similar features are exhibited by the ternary relations $x = \langle y, z \rangle$ and $f`x = y$. Certainly, I can identify $x$ once I know for which pairs $y, z$ it happens that $x = \langle y, z \rangle$ (there is only one!) and if I am given the set of all pairs $f, x$ such that $f`x = y$ then I can recover y. It is this extensionality which enables us to make sense of a notion of *setlike* permutation. Thus, the two operations giving rise to product types (pair) and function types (application) have a kind of extensionality which – as it happens – will enable us to prove a preservation theorem analogous to that in Forster [3]. This point is worth making because the theorem itself will be proved in a kind of generality that does not make its application to these cases blindingly obvious. One reason for this is that it is not customary to think of λ-calculus as a first-order theory in a language with one three-place predicate which reads "x applied to y is z".

This is not the first appearance of permutations in characterising certain kinds of well-typed formulae: Läuchli [6] proves that the intuitionistic theory of the

[1] For an extensive treatment of this technique see Forster [4].

conditional is complete for a certain interpretation that involves permutations. The possibility of a connection between these results has not been explored.

1.2. Possible new directions

What will this preservation theorem do for us? The application of permutation methods to set theory is well, but not widely, understood – one of the minor byways of the subject. The answer is that this theorem raises the possibility that permutation methods could be applied to theories with n-ary extensional relations with the same profit as they have been applied to set theory. In practice, this means λ-calculus, for "f applied to x is y" is a three-place relation extensional in both its second and third arguments and we can express λ-calculus as a first-order theory with equality and this one three-place relation. It may be that permutation methods can shed light on the status of equations between ill-typed λ-terms in the way it has illuminated in NF the status of objects characterised by unstratified formulae such as Quine atoms ($x = \{x\}$), internal $\in$-automorphisms and well-founded sets. To attempt this, we would need a supply of setlike permutations. What setlike permutations can we find in the λ-calculus? The intuitionistic correctness of wffs such as $A \to (B \to C). \leftrightarrow .B \to (A \to C)$ reveals an infinite family of terms like

$$\lambda xyz.xzy$$

all corresponding to setlike permutations. A referee has suggested that the status of dependent types in this context should be discussed. It is not clear that this technique has anything to say about them. As for (infinitary) product types $\prod_\alpha (\alpha \to \alpha) \to \alpha$ and suchlike – these cannot be treated straightforwardly in the same way because these types are not naturally presented as the type of values of some (infinitary) operation in the way that $\alpha \times \beta$ is the type of pairs of things of type α and β. It may be that the most significant consequence of this result is that any theory with one extensional relation generates its own typing scheme.

Since Forster [3] is recapitulated in what follows is not necessary to have read it, though it is a good rehearsal for some of the grimier parts of what is to follow.

We begin by reviewing notation.

2. Definitions

S_X is the group of all permutations of X. $\mathscr{P}$'x is the power set of x. $\mathbf{N}$ is the natural numbers (with S but not $+$ or $\times$), $\mathbf{Z}$ the integers. $j =_{df} \lambda f \lambda x.(f``x)$. A *signature* is an object used to index the sorts of a many-sorted theory. Classically, in a many-sorted theory the syntactic objects – variables, predicate letters etc, all have sort indices, and the structure to which all these indices belong is the *signature*. We will tend to use the letter '$\mathscr{S}$' (to connote both sort and signature) to range over signatures.

We will often use subscripted variables in the style 'x_i'. This is done so that we can define functions on variables *by defining the functions on their subscripts*, so that we do not have to use quotation marks to make apparent that we are talking about variables rather than their values. Sometimes there will be sort-subscripts as well, but they will be s, t not $i, j, k\ldots$. The letters '$i, j, k, \ldots$' will be used as subscripts when we wish to identify members of list of variables $\vec{x}$.

We are assuming here that our sorts are monomorphic, so that, considered as sets, they are formally disjoint. Although languages will have no constants, we shall see later that this is not a serious restriction.

3. Ultrapower lemmas

In [7] Pétry proved an ultrapower lemma for the many-sorted set theory mentioned above. We will need both that and a generalisation to be proved here. We will start with Pétry's ultrapower lemma.

3.1. Petry's ultrapower lemma

$\mathscr{S}$ is to be $\mathbf{N}$, the natural numbers. The sole primitive is $\in$. The following definitions are central.

Definition 3.1. When X is an interval of $\mathbf{Z}$, of a kind to be made precise below, ϕ is a *X-stratified* formula of the language of set theory if we can define on the set of variables occurring free or bound in ϕ an X-valued function σ such that if ϕ contains an occurrence of '$x_i \in x_j$' then $\sigma`j = \sigma`i + 1$ and if ϕ contains an occurrence of '$x_i = x_j$' then $\sigma`i = \sigma`j$. If such a function can be defined on the bound variables we say that ϕ is *weakly X-stratified* and we say the function is a *stratification*.

Although this notion is in principle available for any X that admits a successor function, we will not be interested in finite quotients of the additive group of the positive and negative integers, $\mathbf{Z}$, but only in $\mathbf{N}$, or some interval $\{n: 0 \leqslant n \leqslant k\}$ of it. In the first case we shall call the formulae *stratified* (or *weakly stratified*) and in the second *n-stratified*.

As noted above we will be assuming that the languages we deal with here have no constants. Although this is a simplification it is not a oversimplification: if we wish to prove preservation theorems for a language with constants we would have to recast the proof with "weakly stratified" for "stratified", and the lemmas from Pétry [7] that we use here are proved by him in the appropriate more general form. If there are to be constants, then the appropriate notion of *weakly stratified* allows the stratification to fail to be single-valued on constants.

Definition 3.2. Two extensional structures $\mathscr{M} = \langle M, \in_{\mathscr{M}} \rangle$ and $\mathscr{N} = \langle N, \in_{\mathscr{N}} \rangle$ for the language of set theory are *X-stratimorphic* if there is a family $\{f_i: i \in X\}$ of bijections $M \leftrightarrow N$ indexed by X so that for each i, $\mathscr{M} \models x \in y \leftrightarrow \mathscr{N} \models f_i`x \in f_{i+1}`y$.

We will also need to make frequent use of the ability to switch between a many-sorted and a one-sorted version of a structure. If we have two structures $\langle M, \in_{\mathcal{M}} \rangle$ and $\langle N, \in_{\mathcal{N}} \rangle$ that are elementarily equivalent w.r.t. X-stratified formulae we need two corresponding X-sorted structures that are elementarily equivalent. The natural way to do this is to

Definition 3.3. Obtain from $\mathcal{M} = \langle M, \in_{\mathcal{M}} \rangle$ the structure $\mathcal{M} \times X$ with domain $M \times X$ and a membership relation $\langle m, x \rangle \in_X \langle m', x+1 \rangle$ iff $m \in_{\mathcal{M}} m'$ for each $x \in X$. Let us call this a *sorted version* of $\mathcal{M}$.

Thus, $\langle M, \in_{\mathcal{M}} \rangle$ and $\langle N, \in_{\mathcal{N}} \rangle$ are *strati*morphic iff their sorted versions are *iso*morphic.

X is of course a parameter but we will use this jargon only in cases where it is clear what signature is that we have in mind. The various levels of the sorted version of $\mathcal{M}$ have a natural isomorphism: send $\langle m, x \rangle$ to $\langle m, x+1 \rangle$. Indeed, there is a kind of converse: if an X-sorted structure has an isomorphism that moves the types of its arguments like this then it is a sorted version of a one-sorted structure.

Lemma 3.4. (Petry's ultrapower lemma [7]). *If* $\langle M, \in_{\mathcal{M}} \rangle$ *and* $\langle N, \in_{\mathcal{N}} \rangle$ *are elementarily equivalent w.r.t. stratified sentences they have* $\mathbf{N}$*-stratimorphic ultrapowers.*

Proof. The sorted versions of $\langle M, \in_{\mathcal{M}} \rangle$ and $\langle N, \in_{\mathcal{N}} \rangle$ are elementarily equivalent, by design. Now we invoke a version of Keisler's ultrapower lemma ([2, Theorem 6.1.9]) for many-sorted structures to conclude that the sorted versions of $\langle M, \in_{\mathcal{M}} \rangle$ and $\langle N, \in_{\mathcal{N}} \rangle$ have isomorphic ultrapowers. □

This many-sorted notion of ultrapower is distinct from the one-sorted construction: the signature of the obvious ultrapower is an ultrapower of the signature of the original structure and, thus, may contain nonstandard elements. The ultrapower elements belonging to these nonstandard sorts have to be discarded to leave only objects of standard sorts. The isomorphisms between these two many-sorted stripped-down ultrapowers are precisely stratimorphisms between the corresponding one-sorted structures.

In fact, by saturation, we can show that the ultrapowers will be $\mathbf{Z}$-stratimorphic, though we will not make use of this fact.

3.2. A more general ultrapower lemma

We will be considering languages with one $n+1$-place relation, R, with or without equality. Unless we start off with infinitely many $\geqslant 2$-placed predicates, we can safely suppose that we have precisely one nonlogical many-place predicate, since we can use always glue two predicate letters (an n-place predicate F and an m-place predicate G) together to get (an $n+m$-place) one by

$$H(\vec{x}, \vec{y}) \leftrightarrow_{\mathrm{df}} F(\vec{x}) \wedge G(\vec{y})$$

since then $F(\vec{x}) \leftrightarrow \exists\vec{y}H(\vec{x}, \vec{y})$ and $G(\vec{y}) \leftrightarrow \exists\vec{x}H(\vec{x}, \vec{y})$. This works unless F and G are empty which is not an interesting case.

This is a one-sorted language. We are now going to consider an algebra that will become the signature of a corresponding many-sorted language. An n-ary algebra is an algebra with one n-ary function f. The algebra of interest to us will be the initial n-ary-algebra with countably many generators. Let us call it '$\mathscr{S}$'. The initial 1-ary algebra is of course $\langle \mathbf{N}, S \rangle$; the initial 2-ary algebra is the type algebra of typed λ-calculus.[2]

Definition 3.5. An $n+1$-place relation R is said to be extensional (in its $n+1$th place) iff

$$\forall x, x'(x = x' \leftrightarrow \forall \vec{y}(R(\vec{y}, x') \leftrightarrow R(\vec{y}, x)))$$

and we can define what it is to be extensional in other argument places in the obvious way.

From now on n is a fixed natural number $\geqslant 2$ and $\mathscr{S}$ is the initial $n+1$-ary algebra with countably many generators; the sole operator is f, and $\mathscr{L}$ is the corresponding language. Let ϕ be a formula of $\mathscr{L}$. All our variables are things like 'x_i' for $i \in \mathbf{N}$; so, we can make the following definition:

Definition 3.6. A *minimal* variable is one that never occurs in the $n+1$th place of R in any subformula of ϕ. A *sorting* (of variables) for a formula ϕ of $\mathscr{L}$ is a map μ from the subscripts of the variables (free or bound) in ϕ to $\mathscr{S}$ satisfying:

(1) If '$R(x_1 \ldots x_n, x_{n+1})$' occurs in ϕ then $\mu`(n+1) = f(\mu`1, \ldots \mu`n)$, where the i are the subscripts in '$\vec{x}$'.

(2) If '$x_i = x_j$' occurs in ϕ then $\mu`i = \mu`j$.

Definition 3.7. If there is a sorting for ϕ, ϕ is *sorted*. If there is a function into $\mathscr{S}$ defined only on the indices of the *bound* variables of ϕ but otherwise like a sorting then ϕ is *weakly sorted*.

These are evidently generalisations of the corresponding notions of *stratification* above; in particular, the case $n=2$ corresponds to well-typed formulae of λ-calculus. There will also be analogues of n-stratification, but no suggestive notations, since we

[2] We have restricted attention to languages with one $n+1$-place relation, R, and the corresponding signature we want is the initial algebra with one n-ary relation. In general, we could consider a language with k relations of arities $x_1 \ldots x_k$, and the corresponding signature will be the initial algebra with operations of arities $x_1 - 1 \ldots x_k - 1$. At this stage, the extra complexities of the more general case do not seem to be illuminating, and we will restrict ourself to the one-relation case.

do not have standard ways of denoting initial segments of initial n-ary algebras. Similarly, we need to generalise the notion of *stratimorphic*.

Definition 3.8. $\langle M, R_{\mathcal{M}}\rangle$ and $\langle N, R_{\mathcal{N}}\rangle$ are *$\mathcal{S}$-isomorphic* iff there is a family $\{h_s : s\in\mathcal{S}\}$ of bijections $\langle M, R_{\mathcal{M}}\rangle \leftrightarrow \langle N, R_{\mathcal{N}}\rangle$ such that $\forall \vec{x}\in\mathcal{M}$ and for all n-tuples $\vec{s}\in\mathcal{S}$ and for all $y\in\mathcal{M}$, $\mathcal{M}\models R(\vec{x}, y) \leftrightarrow \mathcal{N}\models R(h_{s_1}`x_1 \ldots h_{s_n}`x_n, h_{f(\vec{s})}`y)$ where 'x_i' is the ith entry in '$\vec{x}$'.

Theorem 3.9. *Let $\mathcal{L}$ be a language with one $n+1$-place primitive R. Let $\mathcal{S}$ be the appropriate n-ary-algebra as above and let Γ be the set of sorted formulae. If $\langle M, R_{\mathcal{M}}\rangle$ and $\langle N, R_{\mathcal{N}}\rangle$ are Γ-elementarily equivalent then $\langle M, R_{\mathcal{M}}\rangle$* and *$\langle N, R_{\mathcal{N}}\rangle$ have $\mathcal{S}$-isomorphic ultrapowers.*

Proof. Exactly as the proof of Lemma 3.4. □

As in the binary case, we can even use saturation to show that the ultrapowers are X-isomorphic where X is the free algebra formed from $\mathcal{S}$ by adding inverses to f (projection functions) f_i for each $i<n$:

$$f_i(f(y)) = y_i,$$

$$x = f(f_1(x), \ldots, f_n(x)).$$

These algebras stand in the same relation to the corresponding initial n-ary algebra as $\mathbf{Z}$ does to $\mathbf{N}$. Johnstone [5] calls these *Jónsson–Tarski* algebras.

4. Rieger–Bernays permutation models

In this section we will prove the two preservation theorems promised earlier. First we consider the model-theoretic construction in its original (ungeneralised) binary version, and then, having used that as an illustration, grind through the generalisation to n-ary languages.

4.1. A preservation theorem for one binary relation

We saw in the introduction the Rieger–Bernays device for proving the independence of the axiom of foundation from ZF. This trick can be applied to any structure with a binary relation (even with equality included – though $(x=y)^\sigma$ is $x=y$) to get a new structure. It will turn out that it is not a sensible move unless the relation is extensional (so, really we are restricted to set theory in this – the simplest – binary case) and the permutation satisfies an extra condition which we must now characterize.

Let $\mathcal{M}$ and $\mathcal{N}$ be structures for the language of set theory which are also models of the axiom of extensionality. Let τ be a bijection $\mathcal{M} \leftrightarrow \mathcal{N}$. Since $\mathcal{M}$ and $\mathcal{N}$ are models of some sort of set theory the map $x \mapsto \tau``x$ gives rise to a map defined on $\mathcal{M}$, since every element of $\mathcal{M}$ corresponds to a unique subset of $\mathcal{M}$. Bearing in mind the definition of j this will be the restriction of $j`\tau$ to $\mathcal{M}$. It may or may not be onto $\mathcal{N}$, for there is, in general, no guarantee that the image of any subset of $\mathcal{M}$ under translation by $\tau$ is a set in the sense of $\mathcal{N}$, which is what we would need for $j`\tau$ (or, strictly, its restriction to $\mathcal{M}$) to be another bijection $\mathcal{M} \leftrightarrow \mathcal{N}$.

Definition 4.1. Fix two structures $\mathcal{M}$ and $\mathcal{N}$. If τ is a bijection $\mathcal{M} \leftrightarrow \mathcal{N}$ such that the restriction of $j^n`\tau$ to $\mathcal{M}$ is onto $\mathcal{N}$ then we say that $\tau$ is *n-setlike*. If $j^m`\tau$ is a bijection $\mathcal{M} \leftrightarrow \mathcal{N}$ for all n then we say that τ is "*setlike*".

(This should be parametrised with the pair $\{\mathcal{M}, \mathcal{N}\}$ but this can be inferred from context.)

The thinking behind this piece of terminology is roughly that we expect the image of a set in a (function which is a) set to turn out to be a set, but we do not (unless we have some form of the axiom scheme of replacement) expect the image of a set in a *class* to turn out to be a set. So, a class such that the image of any set in it is another set is itself in that respect a bit like a set. It is immediate from the definition of setlike that setlike maps are closed under composition and inverses, and from this it follows immediately that the setlike permutations of $\mathcal{M}$ form a subgroup of the symmetric group on $\mathcal{M}$. This definition of "setlike" is the key to the result, for if τ is not setlike we do not have enough control over $\mathcal{M}$ (though we remark without proof that if τ is n-setlike then $\mathcal{M}^\tau$ will satisfy the same $(n+2)$-stratified formulae as $\mathcal{M}$) and on the other hand if τ is a set of $\mathcal{M}$ it may preserve some sentences that are not stratified.

Theorem 4.2. *A sentence in the language of set theory is equivalent to a stratified sentence iff it is preserved in all permutation models where the permutation is setlike.*

The first direction is that permutation models (modulo setlike permutations) preserve stratified sentences. This will be a consequence of the next two observations.

Remark 4.3. *If* $\langle M, \in_{\mathcal{M}} \rangle$ *and* $\langle N, \in_{\mathcal{N}} \rangle$ *are* **N**-*stratimorphic* (*which is the same as there being a setlike bijection* $\mathcal{M} \leftrightarrow \mathcal{N}$) *they satisfy the same stratified sentences.*

Proof. This is an immediate consequence of the definitions. □

Lemma 4.4. *Let* $\langle \mathcal{M}, \in_{\mathcal{M}} \rangle$ *be a structure with equality and one binary extensional relation and let* τ *be a setlike permutation of* $\mathcal{M}$. *Then* $\langle \mathcal{M}, \in_{\mathcal{M}} \rangle$ *and* $\langle \mathcal{M}, \in_{\mathcal{M}} \rangle^\tau$ *are* **N**-*stratimorphic.*

Proof. We will construct $\langle h_i: i\in\mathbf{N}\rangle$ by recursion on i, where each $h_i:\mathcal{M}\to\mathcal{M}^\tau$. h_0 is some arbitrary setlike permutation of $\mathcal{M}$, for the sake of simplicity the identity. Thereafter we will want to know that

$$x\in y \;\leftrightarrow\; h_n\text{`}x\in_\tau h_{n+1}\text{`}y,$$

which is to say

$$x\in y \;\leftrightarrow\; h_n\text{`}x\in\tau h_{n+1}\text{`}y.$$

But we have

$$x\in y \;\leftrightarrow\; h_n\text{`}x\in(j\text{`}h_n)\text{`}y$$

since $u\in v \leftrightarrow \sigma\text{`}u\in(j\text{`}\sigma)\text{`}v$ for any $u$, $v$ and any permutation $\sigma$. So, by extensionality, $\tau h_{n+1}\text{`}y=(j\text{`}h_n)\text{`}y$, which is to say

$$h_{n+1}\text{`}y=(\tau^{-1})(j\text{`}h_n)\text{`}y,$$

i.e. $h_{n+1}=(\tau^{-1})(j\text{`}h_n)$ when restricted to $\mathcal{M}$. So, for $n\geqslant 1$, we set $h_{n+1}$ to be the restriction of $(\tau^{-1})(j\text{`}h_n)$ to $\mathcal{M}$. $\square$

This recursive definition of h_n shows why we have to assume τ is setlike, for, otherwise, the output of the recursion will not be defined on the whole of $\mathcal{M}$.

We shall prove the other direction by appealing to a (slight weakening of a) lemma of Chang and Keisler [2, Lemma 3.2.1, p 124].

Lemma 4.5 (Chang and Keisler [2]). *Let T be a consistent theory in $\mathcal{L}$ and let Δ be a set of sentences of $\mathcal{L}$ which is closed under finite disjunctions and negation. Then the following are equivalent*:

(1) *T has a set of axioms $\subseteq\Delta$.*
(2) *If $\mathcal{A}\models T$ and $\mathcal{A}$ and $\mathcal{B}$ agree on Δ then $\mathcal{B}\models T$.*

We shall let T be a theory with one axiom preserved by setlike permutations. Δ will be the set of stratified wffs. We will need to know that in this case $\mathcal{A}$ and $\mathcal{B}$ have stratimorphic ultrapowers (which is Pétry's ultrapower lemma) and then that if $\mathcal{A}$ and $\mathcal{B}$ are stratimorphic then one is a permutation model of the other.

Lemma 4.6. *$\langle M,\in_{\mathcal{M}}\rangle$ and $\langle N,\in_{\mathcal{N}}\rangle$ are $\mathbf{N}$-stratimorphic iff one is a permutation model of the other (where the permutation is setlike).*

Proof.

$\Rightarrow$: $\langle M,\in_{\mathcal{M}}\rangle$ and $\langle N,\in_{\mathcal{N}}\rangle$ *are stratimorphic; so, for each n*, $(f_{n+1})^{-1}\circ f_n$ is a permutation of $\langle M,\in_{\mathcal{M}}\rangle$. Let us call it h_n. Evidently h_{i+n+1} is the restriction of $j\text{`}h_{i+n}$ to $\mathcal{M}$; so, h_i is n-setlike for each n and, therefore, setlike. $\langle\mathcal{M},\in_{\mathcal{M}}\rangle^{h_n}$ is isomorphic to $\langle N,\in_{\mathcal{N}}\rangle$ for any n: for instance:

$$\langle\mathcal{M},\in_{\mathcal{M}}\rangle^{h_1}\models x\in y$$

iff

$$\langle \mathcal{M}, \in_{\mathcal{M}} \rangle \models x \in h_1 `y$$

iff

$$\langle M, \in_{\mathcal{M}} \rangle \models x \in (f_1)^{-1} \circ f_0 `y$$

iff

$$N \models f_0 `x \in f_1 \circ (f_1)^{-1} \circ f_0 `y = f_0 `y;$$

so, f_0 is an isomorphism between $\langle \mathcal{M}, \in_{\mathcal{M}} \rangle^{h_1}$ and $\langle \mathcal{N}, \in_{\mathcal{M}} \rangle$ and we have shown above that all h_i are setlike.

$\Rightarrow$: For the converse $\mathcal{M}$ and $\mathcal{M}^{\tau}$ are stratimorphic where $h_n = (\tau_n)^{-1}$. τ_n is defined by recursion: τ_0 is the identity and $\tau_{n+1} = \tau \circ j`(\tau_n)$. (Although this notation for τ_n is standard, it is a nonce notation here, in order not to cause confusion with the use of the subscript in h_n.) $\square$

One step forward from here would be to prove the following amplification of an earlier remark, here offered without proof – that the result continues to hold if we replace "stratified" by "$(n+2)$-stratified" and "setlike" by "n-setlike". In the proof we will need the notion of "n-stratimorphic". We could also explore the more general version with constants and weak stratification as developed by Pétry.

However, the reason for presenting this result here was as a rehearsal for the theorem about more general situations in which we encounter sortable formulae, to which we now turn.

4.2. *A more general preservation theorem*

Let R be an $n+1$-place relation extensional in (for the sake of simplicity – it doesn't matter) its $n+1$th place. $n+1$-ary extensional relations on a set X correspond in a natural way to injections $X \hookrightarrow \mathscr{P}`(X^n)$, and it is this characterisation we will need. The injection is $\lambda x \in X.\{\vec{y}: R(\vec{y}, x)\}$. In particular, $\in$ corresponds to the map $\lambda x \in X.(x \cap X): X \hookrightarrow \mathscr{P}`X$. Note that if we input extensional relations into the amalgamation in Section 3.2 we get an extensional relation back. Our definition of a setlike map $X \leftrightarrow Y$ was geared to the extensional relation $\in$. We will now need to generalise it.

Definition 4.7. Let R be an extensional relation on X as above (and S a corresponding relation on Y), we think of R as the map $\lambda x \in X.\{\vec{y}: R(\vec{y}, x)\}: X \hookrightarrow \mathscr{P}`(X)^n$. Let $\sigma_i: i \leqslant n$ be bijections $X \leftrightarrow Y$. Consider a map from X^n to Y^n defined by moving the ith coordinate according to σ_i. This acts naturally (via j) on sets of n-tuples, and thereby on the range of $\lambda x \in X.\{\vec{y}: R(\vec{y}, x)\}$, which is a set of n-tuples.

Notate the resulting map '$J_R(\vec{\sigma})$'. Then if J_R is defined on all n-tuples in the closure of $\{h\}$ under J_R we say that h is *R-setlike* or just *setlike* for short.

So, if $\vec{h}$ is a list of setlike permutations then $J_R(\vec{h})$ is the unique τ such that

$$\forall\vec{x}\forall y\, R(\vec{x}, y) \leftrightarrow R(h_1{}^{\text{‘}}x_1 \ldots h_n{}^{\text{‘}}x_n, \tau{}^{\text{‘}}y).$$

This is the generalisation of *setlike* that we need for the general theorem. As before, the R-setlike maps are closed under composition and inverse, but this is worth spelling out in a little detail.

Lemma 4.8. *Let R_X (R_Y) be a map $X^n \subsetneq \mathscr{P}{}^{\text{‘}}X$ $(Y^n \subsetneq \mathscr{P}{}^{\text{‘}}Y)$ and $\vec{h}(\vec{g})$ a family of bijections $X \to Y$ $(Y \to Z)$. Let* $\mathbf{h}$ $(\mathbf{g})$ *be the map $X^n \to Y^n$ $(Y^n \to Z^n)$ obtained by moving the ith coördinate according to h_i (g_i).*

Then $J_R(\vec{h})$ is $R_Y^{-1}\mathbf{h}R_X$ and $J_R(\vec{g})$ is $R_Z^{-1}\mathbf{g}R_Y$ and $J_R(\vec{gh})$ is the composition of these two, namely $R_Z^{-1}\mathbf{gh}R_X$, which is what we want, and it will be defined if they both are.

This generalises the triviality from the dyadic case, namely that $j{}^{\text{‘}}(\sigma\tau) = (j{}^{\text{‘}}\sigma)(j{}^{\text{‘}}\tau)$. We also need to generalise the idea of a permutation model to the general case:

Definition 4.9.

$$\mathscr{M}^\tau \models R(\vec{x}, y) \leftrightarrow_{\mathrm{df}} \mathscr{M} \models R(\vec{x}, \tau{}^{\text{‘}}y),$$

where R is extensional in the $n+1$th place (occupied by 'y') and $R^\tau(\vec{x}, y)$ is $R(\vec{x}, \tau{}^{\text{‘}}y)$ analogous to $x \in_\tau y$.

The principal result now is Theorem 4.10.

Theorem 4.10. *In a language with one extensional relation R a sentence is sorted iff it is preserved by all permutation models where the permutation is R-setlike.*

The proof of the preservation theorem for permutation models and sorted formulae parallels exactly the proof of the preservation theorem for permutation models and stratified formulae.

First we prove that sorted formulae are preserved by permutation models modulo setlike permutations. Then we show that any two models which agree on sorted formulae have ultrapowers which are $\mathscr{S}$-isomorphic and, therefore, permutation models of each other modulo some setlike permutation, at which point we can invoke Lemma 4.5 again.

Lemma 4.11. *Let $\langle \mathscr{M}, R_{\mathscr{M}} \rangle$ be a structure with equality and one extensional relation and let τ be a setlike permutation of $\mathscr{M}$.* Then *$\langle \mathscr{M}, R_{\mathscr{M}} \rangle$ and $\langle \mathscr{M}, R_{\mathscr{M}} \rangle^\tau$ are $\mathscr{S}$-isomorphic.*

Proof. We will construct $\langle h_s \colon s \in \mathscr{S} \rangle$ by recursion on s, where each $h_s \colon \mathscr{M} \to \mathscr{M}^\tau$. We turn them into components of an $\mathscr{S}$-isomorphism in the obvious way. h_{s_0}, where s_0 is a generator, is some arbitrary setlike permutation of $\mathcal{U}$, for the sake of simplicity of the identity. Thereafter we will want to know that there is $h_{f\vec{s}}$ satisfying

$$\langle M, R_{\mathcal{U}}\rangle \models R(\vec{x}, y) \leftrightarrow \langle M, R_{\mathcal{U}}\rangle^{\tau} \models R(h_{s_1}`x_1 \ldots h_{s_n}`x_n, h_{f\cdot\vec{s}}`y),$$

$$R(\vec{x}) \leftrightarrow R(h_{s_1}`x_1 \ldots h_{f\cdot\vec{s}}`y)^{\tau},$$

where s_i is the sort of the ith variable in '$\vec{x}$' which is to say

$$R(\vec{x}) \leftrightarrow R(h_{s_1}`x_1 \ldots \tau h_{f\vec{s}}`y).$$

But in any case, since h_{s_0} is setlike, we have

$$R(\vec{x}) \leftrightarrow R(h_{s_1}`x_1 \ldots h_{s_n}`x_n, (J_R(h_{s_0} \ldots h_{s_n}))`y);$$

so, by extensionality of R, the desired $h_{f\vec{s}}$ is $\tau^{-1} \circ J_R(h_{s_0} \ldots h_{s_n})$. This map is a composition of two setlike maps and, so, is setlike by Lemma 4.8. □

For the other direction we have to prove that two structures that agree on sorted sentences have ultrapowers that are permutation models of each other. We already have an ultrapower lemma that says they have ultrapowers that are $\mathscr{S}$-isomorphic; so, we need a lemma that says that if two structures are $\mathscr{S}$-isomorphic then one is a permutation model of the other where the permutation is setlike.

Lemma 4.12. *If $\langle M, R_{\mathcal{U}}\rangle$ and $\langle N, R_{\mathcal{V}}\rangle$ are $\mathscr{S}$-isomorphic, one is a permutation model of the other where the permutation is setlike.*

Proof. $\langle M, R_{\mathcal{U}}\rangle$ and $\langle N, R_{\mathcal{V}}\rangle$ are $\mathscr{S}$-isomorphic; so, there is a family $\{h_s\colon s \in \mathscr{S}\}$ such that for any sorts $s_1 \ldots s_n$ and any $x_1 \ldots x_n, y$, there is $h_{f(\vec{s})}$ such that

$$\langle M, R_{\mathcal{U}}\rangle \models R(x_1 \ldots x_n, y) \leftrightarrow \langle N, R_{\mathcal{V}}\rangle \models R(h_{s_1}`x_1 \ldots h_{s_n}`x_n, h_{f(\vec{s})}`y).$$

Clearly, all the h_s are setlike. In particular, we may take all $s_1 \ldots s_n$ to be the same s; so,

$$\langle M, R_{\mathcal{U}}\rangle \models R(\vec{x}, y) \leftrightarrow \langle N, R_{\mathcal{V}}\rangle \models R(h_s`x_1 \ldots h_s`x_n, h_{f(\vec{s})}`y).$$

So,

$$\langle M, R_{\mathcal{U}}\rangle \models R(x_1 \ldots x_{n+1}) \leftrightarrow \langle N, R_{\mathcal{V}}\rangle \models R(h_s`x_1 \ldots h_s`x_n,$$

$$[h_{f(\vec{s})} \circ (h_s)^{-1}] \circ h_s`y),$$

which is simply to say that h_s is an isomorphism between $\langle M, R_{\mathcal{U}}\rangle$ and $\langle N, R_{\mathcal{V}}\rangle^{h_{f(\vec{s})} \circ (h_s)^{-1}}$. Now the permutation $h_{f(\vec{s})} \circ (h_s)^{-1}$ is a composition of two setlike maps and, so, is setlike. □

5. Conclusion

That completes the proof of the preservation theorem for sorted formulae. As indicated, there are various ways in which it could be generalised:

(1) We could consider sorting disciplines for languages with more than one extensional relation. Although the amalgamation in Section 3.2 enables us to replace any

(finite) number of extensional relations by one of very much higher arity, it is probably cheaper to consider instead an initial algebra with a larger number of operations of lower arity. However, the formal definitions of *sorted* and $\mathscr{S}$-*isomorphic* become very complicated with such signatures!

(2) We could prove the corresponding results for languages containing constants, and formulae that are weakly sorted. Some of this is done in the work of Pétry [7].

(3) As hinted earlier, we could prove a sharper version (in the binary case) for formulae that are n-stratified, being those preserved under permutations that are n-setlike, and a corresponding version (in the n-adic case) for formulae that are X-sorted, where X is some fragment of the initial algebra considered in the version proved here.

References

[1] P. Bernays, A system of axiomatic set theory VII, *J. Symbolic Logic* **19** (1954) 86–91.

[2] C.C. Chang and H.J. Keisler, *Model Theory* (North-Holland, Amsterdam, 1974).

[3] T.E. Forster, Permutation models and stratified formulae – preservation theorem, *Z. Math. Logik Grundlag. Math.* **36** (1991) 385–388.

[4] T.E. Forster, *An Essay on Set Theory with a Universal Set*, Oxford Logic Guides (Oxford Univ. Press, Oxford, 1992).

[5] P.T. Johnstone, When is a variety a topos? *Algebra Universalis* **21** (1985) 198–212.

[6] H. Läuchli, An abstract notion of realizability for which intuitionistic predicate logic is complete, J. Myhill et al., eds., *Intuitionism and Proof Theory* (North-Holland, Amsterdam, 1970) 227–234.

[7] A. Pétry, Une caractérisation algébrique des structures satisfaisant les mêmes sentences stratifiées, *Cahiers Centre Logique* **4** (1982) 7–16.

[8] L. Rieger, A contribution to Gödel's axiomatic set theory, *Czechoslovak Math. J.* 7 (82) (1957) 323–357.

[9] D.S. Scott, Quine's individuals, in: Nagel, ed., *Logic, Methodology and Philosophy of Science* (Stanford, 1962) 111–115.