On Free Type Definitions in Z*

R.D. Arthan
ICL Secure Systems
Eskdale Road
Winnersh
Berks. RG11 5TT

Abstract

Recent discussions in the Z community have considered the issue of the consistency of the free type construct in Z. A key question is whether free type definitions which met the criterion for consistency given in the Z Reference Manual, [5], are conservative over Zermelo set theory (i.e. ZF without the axiom of replacement). The main purpose of this paper is to give an introduction to the issues and to show that the answer to this question is "yes" (given the axiom of choice). A by-product of the arguments we give here is that the criterion given in the Z reference manual may be replaced by an intuitively simpler one without loss of expressive power from the theoretical or practical point of view.

1 INTRODUCTION

Z, as described in [5], is a specification language based on classical set theory. An account of a semantics for a significant subset of the language is given in [4] and work on semantics and inference rules is being undertaken as part of an effort to standardise the language.

A Z specification consists of a finite sequence of *paragraphs.* These paragraphs serve to introduce new constants (usually referred to as global variables in the Z literature) and axioms about these constants. In general, it is undecidable whether or not a set of axioms given in a Z specification is consistent. The purpose of this paper is to discuss the consistency of one of the paragraph forms, namely the *free type definition.*

In fact the general theory we develop is quite independent of the Z notation — we take the view that Z is just some language which can be interpreted in first-order set theory. In order to motivate this material, we first discuss consistency issues in general for Z.

The structure of the paper is as follows:

[*] Published in the Proceedings of the 1991 Z User Meeting, Springer Verlag 1992.

- Section 2 discusses some general issues to do with the consistency of specifications and considers some rules for ensuring the consistency of certain forms of specification.

- Section 3 considers some examples in Z intended to illustrate and motivate the theoretical discussion which comprises the remainder of the document.

- Section 4 introduces the formal context in which the results on free type definitions will be proved. The viewpoint here changes from informal considerations about Z to discussing the solutions of certain sorts of inductive definition in first order set theory, and in particular investigating conditions under which such definitions are conservative over Zermelo set theory.

- Section 5 summarises the standard theory in ZF of inductive definitions and introduces terminology for later use.

- Section 6 defines a notion of a typable operator. Since operators arising from Z free type definitions will be typable, and since we shall show that under appropriate conditions such operators are conservative over Zermelo set theory, this notion relates the theoretical treatment to the Z notation.

- Section 7 discusses finitary typable operators and shows that they give rise to conservative inductive definitions. (Note that we use the term "finitary" in the sense which is customary in the theory of inductive definitions. Unfortunately this disagrees with the terminology used in the Z Reference Manual.)

- Section 8 discusses typable operators which satisfy the condition identified in the Z Reference Manual for the consistency of free type definitions. It is shown that such operators are bounded by finitary operators and so that the inductive definitions they give rise to are indeed conservative by the results of section 7.

- Section 9 gives some concluding remarks.

Sections 2 and 3 are largely expository in nature. Readers who are familiar with the issues for Z or whose interest is in the set theoretic results rather than in Z are invited to go directly to section 4, perhaps after a quick perusal of the examples in section 3.

2 Consistency of Z Specifications

It is important when we reason about a specification to be confident that the axioms it contains are consistent, since if they are not then the effort involved in performing proofs is wasted. In this section we set up a simple framework for demonstrating the consistency of certain Z specifications.

The framework consists of some sufficient (but not necessary) conditions for the consistency of certain combinations of Z paragraphs. While this may be of independent interest, it is mainly intended to motivate our reduction of the question of consistency for free type definitions to the simpler and much better understood context of classical set theory. The discussion is informal and many technical details are swept under the carpet, but given any reasonable set of proof rules for Z. it should be possible both to make it rigorous and to extend it to give a set of rules covering most of the specifications which occur in actual practice.

We assume that we have to hand some means for proving assertions of the form $S \vdash p$ where S is a Z specification (i.e. a sequence of Z paragraphs) and p is a predicate which is well-typed with respect to S. S is then said to be *consistent* if it is not possible to prove $S \vdash \mathit{false}$. We assume that there is a notion of a *model* of a specification and that it can be shown that if a specification has a model then it is consistent. In order to give a supply of raw materials for "constructing models", we assume that the mathematical toolkit of [5] is available to us as the specification MT. Finally, let us assume that MT is known to have a model and so to be consistent.

Consider a specification, $MT \frown S$, obtained by appending a list of paragraphs,S, to the mathematical toolkit. In principle, one might hope to demonstrate the consistency of this specification by constructing from S a predicate, C say, which is essentially the conjunction of all the constraints which S places on its global variables, $g1, g2, \ldots$ and then proving an assertion of the form:

$$MT \vdash \exists g1{:}s1;\ g2{:}s2;\ \ldots \mid C$$

If we can prove this then any model for MT must contain elements, $x1, x2, \ldots$ say, which can be used as the interpretations of the global variables, $g1, g2, \ldots$ to give a model for $MT \frown S$ and so $MT \frown S$ must be consistent[1]. However, the construction of C is far from obvious and there are some difficult technical problems. In fact, if S contains both defining and applied occurrences of a generic object, then one cannot hope for a single predicate which will express the consistency of S as a whole in this way. In any case, the consistency assertions would be extremely cumbersome for specifications of any size.

It is sensible, therefore, to look for approaches which let us prove consistency in small steps[2]. Ideally, we would approach the problem one paragraph at a time. This is very natural for axiomatic definitions for which we have the following:

Consistency Rule for Axiomatic Definitions Consider an axiomatic definition:

[1] A stronger statement is true: S is a *conservative extension* of MT, that is to say any model of MT can be expanded to a model of $MT \frown S$ without actually adding any new sets. S just gives new names for things which must already exist in a model for MT. Restricting oneself to conservative extensions of MT does not have any significant impact on the power of Z, since MT gives us the same mathematical power as first-order Zermelo set theory and, to quote an exercise in [2], "99% of mathematics can be carried out in Zermelo set theory". It is the purpose of this paper to show that free type definitions in Z do not live in the missing 1%!.

[2] Indeed, an approach in which consistency is demonstrated in small steps, each of which demonstrates the conservativeness of a small number of paragraphs with respect to what has gone before, seems to be the only practical way of handling the consistency problem which allows the proof obligations to be stated directly in Z. The alternative of resorting to proof obligations couched in terms of meta-linguistic statements about the specification viewed as a syntactic object would be intractably complex.

z

$c1 : s1;\ c2 : s2;\ ...$

p

If this definition, say A, is well-typed with respect to the consistent specification S and if we can prove the assertion:

$S \vdash \exists c1 : s1;\ c2 : s2;\ ... \mid p$

then the specification $S^\frown\langle A\rangle$ is consistent. □

Provided we are happy to introduce all the constraints which we plan to impose on the c_i in one go, the above rule (which is implicit in [4, section 5.2]) gives a useful approach to the consistency problem.

It is actually straightforward to extend the above approach to give rules for all of the other Z paragraph forms except given set, free type definitions and the constraint paragraph (in which one just writes down an arbitrary axiom). (Generic boxes are exactly like axiomatic definitions except that the assertion we have to prove is a generic one; other forms such as schema boxes and abbreviation definitions are always consistent).

The free type paragraph form is explained in [5] as equivalent to a given set paragraph immediately followed by an axiomatic definition. This is a special case of a common Z idiom in which a new type is introduced using a given set paragraph and then axioms are introduced which populate the given set (or constrain its population) over several paragraphs. We offer the following rule to handle the special case of interest:

Consistency Rule for Given Sets Consider the following fragment of specification comprising a given set declaration followed by an axiomatic definition

$[X]$

z

$c1 : s1;\ c2 : s2;\ ...$

p

If this fragment, say GA, is well-typed with respect to the consistent specification S and if for some set-valued term t we can prove the assertion:

$S \vdash (\exists c1 : s1;\ c2 : s2;\ ... \mid p)[t/X]$

where $q[t/X]$ denotes the result of substituting t for X in q, then the specification $S^\frown GA$ is consistent. □

Note that the above rule assumes, for simplicity, that given sets are allowed to be empty. If you want to exclude this, e.g. to make the proof rules simpler, then just add a clause to assert that t is not empty. We will work with the slightly simpler formulation which allows empty given sets.

The Z Reference Manual [5], gives a condition for the consistency of a free type definition. For reference we give a slightly reworked form of this condition:

ZRM Rule for Free Types Consider the free type definition:

$$T ::= c_1 \mid c_2 \mid ... \mid d_1\langle\langle E_1\rangle\rangle \mid d_2\langle\langle E_2\rangle\rangle \; ...$$

If this definition, say F, is well-typed with respect to the consistent specification S and if for each term E_c we can prove the following assertion (which is generic in X):

$$\begin{array}{l} S \vdash \forall S{:}seq(\mathbb{P}X) \mid \forall i{:}\mathbb{N}\bullet S(i) \subseteq S(i{+}1) \bullet \\ \qquad \bigcup\{i{:}\mathbb{N}\bullet E_c[S(i)/T]\} = E_c(\bigcup\{i{:}\mathbb{N}\bullet S(i)]\}) \end{array}$$

then the specification $S\frown\langle F\rangle$ is consistent. □

The motivation for this rule is discussed in [5] and in the remark following example 6 in the following section.

3 Some Examples

In this section we put the consistency rules of the previous section to work on some examples of free type definitions. Further examples are considered in [4, section 5.2] and in [3].

Example 1 The simplest examples of a free type definition are the ones in which none of the constructors have arguments. For example:

z

$$DIRECTION ::= N \mid S \mid E \mid W$$

Following the recipe in [5], we find that this is equivalent to:

z

$$[DIRECTION]$$

z

$$\begin{array}{l} N,\ S,\ E,\ W : DIRECTION \\ \hline disjoint\langle\{N\}, \{S\}, \{E\}, \{W\}\rangle \\ \land\ \forall X{:}\mathbb{P}\ DIRECTION\bullet\{N, S, E, W\} \subseteq X \Rightarrow DIRECTION \subseteq X \end{array}$$

Applying our consistency rule for given sets, this is consistent with respect to the mathematical toolkit if we can find a set-valued term t such that:

$$\begin{array}{l} MT \vdash \exists N,\ S,\ E,\ W : t \mid \\ \qquad disjoint\langle\{N\},\ \{S\},\ \{E\},\ \{W\}\rangle \\ \quad \land\ \ \forall X{:}\mathbb{P}\ t \bullet \{N,\ S,\ E,\ W\}\ \subseteq X \Rightarrow t \subseteq X \end{array}$$

Now, we can reasonably expect to be able to prove the above assertion by taking t to be a set containing precisely four integers, such as $\{\mathit{0}, \mathit{1}, \mathit{2}, \mathit{3}\}$. Having proved this we have established the consistency of the original free type definition.

Example 2 This example illustrates a useful little trick. Consider the free type definition:

$$XORY ::= InX\ \langle\langle X\rangle\rangle \mid InY\ \langle\langle Y\rangle\rangle$$

This is well-typed with respect to the specification $MT ^\frown \langle[X], [Y]\rangle$ consisting of the mathematical toolkit and two given set definitions and says that $XORY$ is the disjoint union of X and Y. Our consistency rule asks us to find a t such that we can prove:

$$\begin{array}{l} MT\ ^\frown\ \langle[X],\ [Y]\rangle \vdash \exists InX : X \rightarrowtail t;\ InY : Y \rightarrowtail t \mid \\ \qquad\qquad disjoint\langle\{ran\ \ InX\},\ \{ran\ \ InY\}\rangle \\ \qquad \land\ \ \forall W{:}\mathbb{P}\ t \bullet InX(\!|X|\!) \cup InY(\!|Y|\!)\ \subseteq W \Rightarrow t \subseteq W \end{array}$$

One's first guess at such a t is to use $X \times Y \times \{\mathit{0}, \mathit{1}\}$ using the last component to indicate which side of the disjoint union an element is in, but this fails since, for example, given $x \in X$ we do not have to hand a convenient element of Y to serve as the second component of $InX(x)$. The trick is to use $\mathbb{P}X \times \mathbb{P}Y \times \{\mathit{0}, \mathit{1}\}$ and take InX and InY defined by

$$\begin{array}{l} InX\ \ x = (\{x\},\ Y,\ \mathit{0}) \\ InY\ \ y = (X,\ \{y\},\ \mathit{1}) \end{array}$$

We leave it to the interested reader to check that this works.

Remark It is not hard to see that the techniques of example 1 and 2 generalise straightforwardly to show that any free type definition which is not recursive is consistent. Thus all the interest is in the recursive examples.

Example 3 The simplest example of a consistent recursive type definition is the rather trivial:

$$TRIV ::= Triv\ \langle\langle TRIV\rangle\rangle$$

Since we have not imposed the restriction that given sets be non-empty, it is not hard to see that this is consistent.

Example 4 The following recursive type definition is much more interesting:

$$T ::= u \ \langle\!\langle \mathbb{F}T \rangle\!\rangle$$

We can think of T as the set of all finitely branching trees in which the order of the branches at a node is immaterial and in which all the leaves are identical (namely they are all $u(\varnothing)$. The consistency rule challenges us to find a t such that:

$$MT \vdash \exists u : \mathbb{F}t \rightarrowtail t \mid \forall W{:}\mathbb{P}\ t\bullet u\llparenthesis \mathbb{F}W \rrparenthesis \ \subseteq\ W \Rightarrow t \subseteq W \qquad (*)$$

A useful strategy in finding such a t is to break the problem down into two parts: first of all find an s which admits an injection $w : \mathbb{F}s \rightarrow s$ (i.e. in a certain sense s is closed under the formation of finite subsets of its elements); then extract a suitable t from s by forming the intersection of all the w-closed subsets of s. I.e. take

$$t = \bigcap \ \{a : \mathbb{P}s \mid w\llparenthesis \mathbb{F}a \rrparenthesis \subseteq a\}$$

For then, taking $u = \mathbb{F}t \lhd w$ to be the restriction of w to the finite subsets of t, it is not hard to show that t satisfies $(*)$ above. The hard part of the proof is showing that $w\llparenthesis \mathbb{F}t \rrparenthesis \subseteq t$, which amounts to showing that $w\llparenthesis \mathbb{F}t \rrparenthesis \subseteq a$ for any w-closed subset a of u; but for such an a, we must have $a \subseteq t$, whence $w\llparenthesis \mathbb{F}t \rrparenthesis \subseteq w\llparenthesis \mathbb{F}a \rrparenthesis \subseteq a$, as required, since the relational image operation is *monotonic*, i.e. if $A \subseteq B$ then $R\llparenthesis A \rrparenthesis \subseteq R\llparenthesis B \rrparenthesis$. This proof is essentially that of the Knaster-Tarski fixed-point theorem.

The above discussion reduces the proof of $(*)$ to finding a set s and an injection $w : \mathbb{F}s \rightarrow s$. A traditional answer is to take the s to be $\mathbb{N}$ and w to be the function which sends a set $\{x_1, x_2, \ldots, x_k\}$ to $2^{x_1} + 2^{x_2} + \ldots + 2^{x_k}$, i.e. the number whose representation in binary has a 1 in positions $x_1, x_2, \ldots, x_k$. It is not hard to show that this does actually define an injection and so enables us to prove $(*)$.

Remark The strategy discussed above, in which we first find a set satisfying a closure property determined by the free type definition in question and then form the free type as a minimal closed subset is quite general and saves us having to reason about the induction property for the free type since the minimal closed subset has the induction property automatically.

Example 5 The consistency of the following free type definitions is not hard to demonstrate:

$$\begin{aligned} &FR ::= \mathit{fr}\ \langle\!\langle \mathbb{F}(\mathbb{N} \times FR) \rangle\!\rangle \\ &FF ::= \mathit{ff}\ \langle\!\langle \mathbb{F}(\mathbb{N} \mathrel{⇻} FF) \rangle\!\rangle \end{aligned}$$

(For *(i)* one can use a bijection, say $f : \mathbb{N} \times \mathbb{N} \mathrel{⤖} \mathbb{N}$ and the previous example to construct a bijection $g : \mathbb{F}(\mathbb{N} \times \mathbb{N}) \mathrel{⤖} \mathbb{N}$; For *(ii)* one uses FR and the fact that $\mathbb{F}(\mathbb{N} \mathrel{⇻} X) \subseteq \mathbb{F}(\mathbb{N} \times X)$ for any X.)

Remark Examples 4 and 5 show the consistency of simple cases of recursive free types involving only $\mathbb{N}$[3], $\times$ and $\mathbb{F}$. This generalises and, in fact, all recursive free types involving only these things are consistent. Moreover, the free types in question have a simple finite-tree-like structure: their elements may be formed by starting from nothing, and repeatedly applying the constructor function to what has already been constructed. E.g. in example 4, we may construct the free type as the union of the following sets X_i:

$$\begin{array}{|l} T_0 = \varnothing \\ T_1 = u(\mathbb{F}\, T_0) = \{u(\varnothing)\} \\ T_2 = u(\mathbb{F}\, T_1) = \{u(\varnothing),\ u(\{u(\varnothing)\})\} \\ \ldots \end{array}$$

Thus a simple computable process will eventually generate any element of the free type. However, there are free types, which are consistent by the ZRM rule of consistency for free type definitions, but which are much too big too be formed by any computable process, as the following example shows.

Example 6 The following free type definition may be checked to satisfy the ZRM rule.

$$\begin{array}{|l} Y ::= y \langle\!\langle\ \{r{:}\mathbb{N} \leftrightarrow Y \mid ran\ r \in \mathbb{F}\, Y\}\ \rangle\!\rangle \end{array}$$

This differs from the earlier example *FR* in that, whereas *FR* was defined by recursion using finite relations, we now use relations which may be infinite provided they only contain finitely many elements of Y.

It is not trivial to work out how big the set Y must actually be. We can construct Y in stages as discussed above:

$$\begin{array}{|l} Y_0 = \varnothing \\ Y_1 = \{y\{r{:}\mathbb{N} \leftrightarrow \varnothing \mid ran\ r \in \mathbb{F}\varnothing\}\} = \{y(\varnothing)\} \\ Y_2 = \{y\{r{:}\mathbb{N} \leftrightarrow \{fr(\varnothing)\} \mid ran\ r \in \mathbb{F}\{y(\varnothing)\}\}\} = \mathbb{P}(\mathbb{N} \times \{y(\varnothing)\}) \\ \ldots \end{array}$$

So that Y_2 is already uncountably infinite, since $\mathbb{P}(\mathbb{N} \times \{y(\varnothing)\})$ is in 1-1 correspondence with the uncountable set $\mathbb{P}\mathbb{N}$.

Remark The reason why the ZRM rule does guarantee consistency may be seen in terms of this example, as follows. One may check that $Y_i \subseteq Y_{i+1}$ for each i. It is then fairly easy to see that the condition imposed by the rule implies that the union of the Y_i satisfies the closure conditions required by the free type definition. The snag with this argument is that the construction cannot be carried out within Z because formation of this union violates the Z typing rules.

The main question this paper aims to address is whether or not free type definitions which are known to be consistent by the ZRM rule are actually conservative

[3] For present purposes I could equally well have used $\mathbb{Z}$, since only the fact that $\mathbb{N}$ is countably infinite is relevant.

extensions, i.e., effectively, whether a construction of the free type is possible within the world of sets we can construct using the mathematical tool-kit. This world of sets comprises any subset of any type we can construct starting from the type of integers using the primitive type-forming operators of Z.

The next example shows that this would not be so if we abandoned the Z type discipline.

Example 7 Consider the following ill-typed attempt at a Z free type definition:

$$V ::= v\langle\!\langle\ \{\mathbb{N}\} \cup V \cup \{x : V \bullet \mathbb{P}x\} \cup \bigcup V\ \rangle\!\rangle$$

What this says is that V is in 1-1 correspondence with a set comprising *(a)* the set of natural numbers, *(b)* the elements of V, *(c)* the powerset of any element of V, and *(d)* any element of an element of V. Perhaps surprisingly, this construction satisfies the ZRM condition. To see that such a set is just too large, one considers the construction of a sequence of sets whose union would be V. What happens may be seen in outline as follows:

$$\begin{aligned}
&V_{\mathit{0}} = \varnothing \\
&V_{\mathit{1}} = \{\mathbb{N}\} \\
&V_{\mathit{2}} = v(\ldots \cup \{x : \{\mathbb{N}\} \bullet \mathbb{P}x\} \cup \textstyle\bigcup\{\mathbb{N}\}) && = v(\ldots \cup \{\mathbb{P}\mathbb{N}\} \cup \mathbb{N}) \\
&V_{\mathit{3}} = v(\ldots \cup \{x : V_{\mathit{2}} \bullet \mathbb{P}x\} \cup \textstyle\bigcup\{\mathbb{P}\mathbb{N}\}) && = v(\ldots \cup \{\mathbb{P}\mathbb{P}\mathbb{N}\} \cup \mathbb{P}\mathbb{N} \cup \mathbb{N} \cup \ldots) \\
&\ldots
\end{aligned}$$

At stage *1*, $\mathbb{N}$ appears as an element, at stage 2, $\mathbb{P}\mathbb{N}$ appears as an element and $\mathbb{N}$ appears as a subset, at stage 3, $\mathbb{P}\mathbb{P}\mathbb{N}$ appears as an element and $\mathbb{P}\mathbb{N}$ and $\mathbb{N}$ are subsets, and so on. Thus for any k V must contain a subset in 1-1 correspondence with $\mathbb{P}^k\mathbb{N}$ and this is impossible for any set whose type is build up from the integers using the Z type constructors.

Remark The set-theoretic axiom which is used to justify the formation of sets like the union of the V_i in the above example is the axiom of replacement. This axiom may be used to justify the ZRM rule for free type definitions. Unfortunately, there is no satisfactory way to express this axiom in a typed language like Z. This is closely related to the fact that the main effect on set theory of the axiom of replacement is to admit very large sets (such as the set V discussed above). The main result proved in the sequel is that the ZRM rule may equally well be justified using the axiom of choice. The choice axiom has the advantage that it may naturally be stated in Z, as we see in the final example.

Example 8 The axiom of choice may be stated in Z as the following assertion, generic in X and Y.

$$\vdash (\exists f{:}X \twoheadrightarrow Y \bullet \mathit{true}) \Rightarrow (\exists g{:}Y \rightarrowtail X \bullet \mathit{true})$$

That is to say, for any sets X and Y, of any type, if there exists a surjection from X to Y, then there exists an injection from Y to X.

4 Reduction To Set Theory

For the purposes of the rest of this paper, we are not much concerned with the actual syntax of the Z notation. Our point of view will be that the Z is just some language with an interpretation in first-order set theory.

Z is actually a typed (or many-sorted) language. For this paper, the import of this is a restriction on the sort of first-order assertions one can make in Z. This restriction is stated as a general property of first order sentences in section 6 below.

Z has various extension mechanisms, i.e. means of introducing new logical constants[4] into the language. We are currently concerned with the mechanism for introducing the so-called *free types*.

The form of a Z free type definition is[5]:

$$T ::= c_1 \langle\langle e_1 \rangle\rangle \mid c_2 \langle\langle e_2 \rangle\rangle \mid \ldots \tag{1}$$

where T is the new type being defined and the e_i are set-valued expressions (which may involve T).

The intention of (1) is that a new set-valued constant T should be introduced together with functions c_i displaying T as the disjoint union of the sets e_i. That is to say, *(i)* each c_i is an injection of e_i into T, *(ii)* the ranges of the c_i are pairwise disjoint, *(iii)* T is contained in the union of these ranges.

The Z reference manual also imposes the additional restriction, that, *(iv)* no proper subset of T is closed under the c_i, i.e., the following induction principle holds for the set T: "*if $W \subseteq T$ and for each i, the image of $e_i[W/T]$ under c_i is contained in W, then $W = T$*" (where $e_i[W/T]$ denotes the result of substituting W for T in the expression e_i). As we remark after lemma 2 below, it turns out to be possible to construct a T satisfying *(iv)* given one satisfying only conditions *(i)*, *(ii)* and *(iii)*, so let us agree to ignore *(iv)* for the time being.

In the terminology of first order set-theory, we may think of a Z free type definition as simply an "equation" of the form $T \simeq \phi(T)$, where the operator ϕ is the expression formed from the right hand side of a free type definition such as (1) above by deleting the c_i and the chevrons and by replacing the |s with a suitable disjoint union operator. Here $A \simeq B$ is intended to mean that there is a bijection between A and B.

Thus the set-theoretic formulation of (1) is just

$$T \simeq \phi(T) \tag{2}$$

and the question is: when do equations of the form (2) have solutions T in whatever set theory we think of as underlying Z?

[4] The Z literature uses the term "global variable" rather than "constant".

[5] In fact Z allows the more general syntax:

$$T ::= d_1 | d_2 | \ldots | c_1 \langle\langle e_1 \rangle\rangle | c_2 \langle\langle e_2 \rangle\rangle | \ldots$$

Here the "nullary constructors" d_i are intended to be distinct members of the new type T. For simplicity, we treat such a constructor as a c_i with the corresponding e_i a singleton set.

I believe that Z is usually thought of as being based on Zermelo set theory (i.e. ZF, but without the replacement axiom). This is the line taken in the treatment of the semantics of Z in [4] and it is the line we shall follow here. It is a natural choice, since the universe of types in Z is constructed from ground types by finite iterations of power-set and product operations. Thus, if we start from a countably infinite type we end up with a universe of types whose union is precisely the standard universe for Zermelo set theory.

5 Inductive Definitions in ZF

In general we cannot expect equations such as (2) to have a solution at all. The simplest example is when $\phi(X) = \mathbb{P}(X)$. For in that case (2) would conflict with a famous result of Cantor. Conditions under which such an equation does have a solution are considered in, for example, the section on Inductive Definitions by P. Aczel in [1]. The case discussed there is actually somewhat different from ours, in that Aczel is concerned with solutions to the equation $\phi(X) = X$ "on the nose" rather than up to a bijection, and he also is interested in the case where the domain and range of ϕ are contained in the power set of some pre-existing set. In this case, it is sufficient for ϕ to be monotonic, in the sense that whenever $A \subseteq B$, then $\phi(A) \subseteq \phi(B)$. Indeed, there are two well-known methods of constructing a fixed point for a monotonic function $\phi : \mathbb{P}(X) \to \mathbb{P}(X)$.

The first method constructs the fixed point "from below" by starting from the empty set and then transfinitely iterating the ϕ-closure operation, $A \mapsto A \cup \phi(A)$[6]. More formally, we define a sequence ϕ_α of sets by (transfinite) induction:

$$\phi_\alpha = (\bigcup_{\beta<\alpha} \phi_\beta) \cup \phi(\bigcup_{\beta<\alpha} \phi_\beta) \tag{3}$$

Since $\phi_\alpha \subseteq X$ for any ordinal α, the union of the ϕ_α may be formed (without recourse to the axiom of replacement) and may readily be seen to be a fixed point of ϕ.

The second method constructs the fixed point "from above", as the intersection of all ϕ-closed sets, where a set is ϕ-closed if it contains the ϕ-closure of any of its subsets[7].

Unfortunately, in our context we have no pre-existing set which is ϕ-closed and so even for monotonic ϕ neither construction can immediately be applied. Given an arbitrary term, ϕ, of ZF, with the variable X free, we can attempt to perform the construction from below, but the construction will not in general converge, even for monotonic ϕ, and so may fail to produce a fixed point for ϕ. The existence of a ϕ-closed set for monotonic ϕ is sufficient for convergence (and the proof does not use the axiom of replacement, see lemma 1).

A sufficient criterion for convergence in ZF is a condition referred to as "finitariness" in [5]. The condition on ϕ is that whenever $X_1 \subseteq X_2 \subseteq \ldots$ is a countable chain of sets, then $\phi(\bigcup X_i) = \bigcup \phi(X_i)$. It is easy to see that such ϕ are monotonic

[6] Here and throughout the sequel, I use the usual mathematical notation in which $x \mapsto f(x)$ denotes the function $\lambda x.f(x)$.

[7] Note that if ϕ is monotonic, A is ϕ-closed iff. $\phi(A) \subseteq A$ (see lemma 1).

(since $A \subseteq B$ iff. $A \cup B = B$) and that the construction of a fixed point from below converges at or before ω, the first infinite ordinal. Note, however, that we need the axiom of replacement to assert the existence of ϕ_α for infinite α. Thus this condition is not on its own sufficient to supply a fixed point in Zermelo set theory. An explicit counterexample is given in section 6 below.

Unfortunately, Aczel also uses the term finitary for a stronger (and more natural) condition than the one given in [5]. To avoid confusion, let us fix some terminology for the rest of the paper:

- an *operator* is a term, ϕ, or, for emphasis, $\phi(X)$, in the language of first order set theory in which X may appear free[8];
- an operator ϕ is *monotonic* if it satisfies the condition:

$$\forall X \bullet \forall Y \bullet Y \subseteq X \Rightarrow \phi(Y) \subseteq \phi(X);$$

- if ϕ is an operator, a set X is *ϕ-closed* if $\phi(A) \subseteq X$ for every $A \subseteq X$.
- an operator ϕ is *cts*[9] if it commutes with countable unions of chains as discussed above, i.e. for any chain $X_1 \subseteq X_2 \subseteq \ldots$, we have:

$$\phi(\bigcup_i X_i) = \bigcup_i \phi(X_i);$$

- finally, an operator ϕ is *finitary*, if it satisfies the condition:

$$\forall X \bullet \phi(X) = \bigcup\{\phi(Y) | Y \subseteq X \wedge Y \text{ is finite}\}.$$

The above use of *finitary* agrees with Aczel. Finitary operators are easily seen to be cts.

The following lemma records some of the facts which have been mentioned in passing in this section for future reference.

Lemma 1 *Let ϕ be a monotonic operator, then*

(i) *a set X is ϕ-closed if $\phi(X) \subseteq X$.*

(ii) *if a set X is ϕ-closed, then for some $A \subseteq X$, $\phi(A) = A$.*

Proof: *(i)* Note that ϕ-closure is a condition on the behaviour of ϕ on the *subsets* of X not on its elements — what we have to prove is that if $\phi(X) \subseteq X$ then

[8] To be more formal, I should work with two place predicates, $\Phi(Y, X)$ say, which are functional in Y and then use appropriate circumlocutions for assertions involving ϕ. For example, $\phi(A) = B$ would be viewed as an abbreviation for $\Phi(A, B)$. Standard technical devices exist to justify the terminology with ϕ, see e.g [2].

[9] I.e. *C*ondition *t*aken from *S*pivey, or, if you must, continuous with respect to the topology on the ZF universe in which the open sets are the sets which are inaccessible from countable directed unions. To make the second interpretation precise would require us to make precise the idea of a "topology" in which the open sets were actually proper classes.

$\phi(A) \subseteq X$ for any $A \subseteq X$. But this follows from monotonicity, since, if $A \subseteq X$ then $\phi(A) \subseteq \phi(X) \subseteq X$, so $\phi(A) \subseteq X$ as required.

(ii) If X is ϕ-closed then ϕ restricts to a monotonic function, $\psi : \mathbb{P}(X) \rightarrow \mathbb{P}(X)$ say. A fixed point for ψ (and so for ϕ) may be found using either of the two methods discussed above. □

Like all of the results in this paper, lemma 1 is actually a metatheorem and, for example, part *(i)* would more precisely be stated as "*For any operator ϕ, it is a provable in Zermelo set theory that 'ϕ is monotonic implies that a set X is ϕ-closed if $\phi(X) \subseteq X$*'". Even more rigorously, the statement would be "*For any operator ϕ, it is a provable in Zermelo set theory that '$(\forall X \bullet \forall Y \bullet (Y \subseteq X \Rightarrow \phi(Y) \subseteq \phi(X))) \Rightarrow \forall X \bullet (\phi(X) \subseteq X \Rightarrow \forall A \bullet (A \subseteq X) \Rightarrow \phi(A) \subseteq X$*'". We will continue to use the less formal terminology.

The following lemma is the analogue of part *(ii)* of lemma 1 for fixed points up to bijection. It turns out to be the only means we will need to construct such fixed points.

Lemma 2 *Let ϕ be a monotonic operator and X a set such that there is an injection $f : \phi(X) \rightarrow X$, then for some $Y \subseteq X$, $\phi(Y) \simeq Y$.*

Proof: Define an operator ψ by:

$$\psi(A) = \{x : X | \exists y : \phi(A) \bullet f(y) = x\} \tag{4}$$

i.e. $\psi(A)$ is the image of $\phi(A)$ under f. Then it is easy to check that ψ is monotonic and that X is ψ-closed, and so, by lemma 1, ψ has a fixed point $Y \subseteq X$. But then

$$Y = \psi(Y) = \{x : X | \exists y : \phi(Y) \bullet f(y) = x\} \simeq \phi(Y) \tag{5}$$

since f is an injection. □

In section 4 we mentioned that the fixed points that are needed for Z free type definitions are actually required to satisfy an additional induction principle. It is straightforward to prove by (transfinite) induction that the construction of a fixed point from below produces a least fixed point. This implies that we may choose to pick Y in the above proof so that if $B \subseteq X$ satisfies $\psi(B) = B$, then $Y \subseteq B$. It is not hard to derive from this an induction principle for ψ which translates into the desired induction principle. Alternatively, after applying the above lemma as it stands to give a bijection say $g : Y \rightarrow \phi(Y)$, the intersection of all $(g^{-1} \circ \phi)$-closed subsets of Y will also be a fixed point up to a bijection and will satisfy the induction principle.

6 Typable Operators

The operators in which we are interested arise from from well-typed Z free type definitions. Thus, in a sense we shall now explain, we may restrict our attention to operators which are typable. The types we will assign to our operators will come from the language of types[10]given by the following clauses: parameter:

[10] This is not the type system of Z as in [4], which includes arbitrary n-tuple types and labelled record types (called schema types). Since semantics for n-tuple and labelled record types can be

1. The variables X and ω are types
2. If τ_1 and τ_2 are types then so are $(\tau_1 \times \tau_2)$, $(\tau_1 \sqcup \tau_2)$, and $(\mathbb{P}\tau_1)$.
3. Nothing is a type except by virtue of rule 1 or rule 2.

Given a set Y and a type τ, we may evaluate the type τ at Y, to give what we shall write as $\tau(Y)$, using the following rules:

$$\begin{aligned} X(Y) &= Y && (6)\\ \omega(Y) &= \omega && (7)\\ (\tau_1 \times \tau_2)(Y) &= \tau_1(Y) \times \tau_2(Y) && (8)\\ (\tau_1 \sqcup \tau_2)(Y) &= \tau_1(Y) \sqcup \tau_2(Y) && (9)\\ (\mathbb{P}\tau)(Y) &= \mathbb{P}(\tau(Y)) && (10) \end{aligned}$$

where ω is the set of natural numbers, $\mathbb{P}$ is the usual power set operator, × is binary cartesian product, and $\sqcup$ is binary disjoint union. Note that the variable X acts as a formal parameter of the type τ. I assume here that some fixed functorial construction of products and disjoint unions has been chosen — the usual ones will do nicely (with $A \sqcup B$ a subset of $(A \cup B) \times \{0, 1\}$). It will later on be important that the product and disjoint unions we use are constructible in Zermelo set theory (i.e. without the axiom of replacement) — again the usual constructions are adequate.

Note that for any type τ, $Y \mapsto \tau(Y)$ is the objects part of a functor from sets to sets. The morphisms part of the functor is given by $(f : X \to Y) \mapsto (\hat{\tau}(f) : \tau(X) \to \tau(Y))$ where $\hat{\tau}(f)$ is defined by the following rules:

$$\begin{aligned} \hat{X}(f) &= f && (11)\\ \hat{\omega}(f) &= n \mapsto n && (12)\\ (\tau_1 \hat{\times} \tau_2)(f) &= (x_1, x_2) \mapsto ((\hat{\tau_1}(f))(x_1), (\hat{\tau_2}(f))(x_2)) && (13)\\ (\tau_1 \hat{\sqcup} \tau_2)(f) &= \iota_j(x) \mapsto (\hat{\tau_j}(f))(x) && (14)\\ (\widehat{\mathbb{P}\tau})(f) &= s \mapsto \{y : \tau(Y) | \exists x \bullet (\hat{\tau}(f))(x) = y\} && (15) \end{aligned}$$

where the ι_j $(j = 1, 2)$ are the injections of the summands into the disjoint union.

Clearly for any τ, $X \mapsto \tau(X)$ is monotonic and $(f : X \to Y) \mapsto (\hat{\tau}(f) : \tau(X) \to \tau(Y)$ preserves injections, surjections and bijections (by an easy induction over the structure of τ). We shall make no further use of categorical language except for occasionally observing that $X \mapsto \tau(X)$ is a functor which preserves injections etc.

given in terms of the binary product type we use here, our discussion should cover the apparently more general types of Z.

We have only allowed a single countably infinite ground type, ω, and some of our arguments will use the fact that the ground types are all infinite to avoid having special cases for finite ground types. The general case may be handled either by reformulating our results to handle additional finite ground types, or, probably better, by observing that additional ground types introduced by *conservative* extensions may be treated semantically within the framework of types we have given here.

Now let us say that an operator, $\phi(X)$, is *typable* with type τ iff. we can prove $\forall X \bullet \phi(X) \subseteq \tau(X)$. E.g. the following operators may be seen to be typable:

$$\begin{aligned} \phi_{\mathit{1}}(X) &= \mathbb{P}(X) && (16)\\ \phi_{\mathit{2}}(X) &= X \sqcup \mathit{1} && (17)\\ \phi_{\mathit{3}}(X) &= (\omega \times X) \sqcup \mathit{1} && (18)\\ \phi_{\mathit{4}}(X) &= \{x : X | X \text{ is uncountable}\} && (19) \end{aligned}$$

where $\mathit{1}$ is a singleton set.

All of the above operators are monotonic. ϕ_{1} is the example we have already mentioned of an operator which has no fixed point (even up to a bijection). ϕ_{2} and ϕ_{3} correspond to free types representing the natural numbers and lists of elements of ω respectively. ϕ_{4} is an example of an operator which is cts but not finitary. $\phi_{\mathit{4}}(X)$ is X, if X is uncountable, and empty otherwise. ϕ_{4} has type X and its fixed points are the empty set and the uncountable sets.

Some examples of operators which are not typable are:

$$\begin{aligned} \phi_{\mathit{5}}(X) &= \bigcup X && (20)\\ \phi_{\mathit{6}}(X) &= \{x : (\mathbb{P}(\bigcup X)) \cup \{\omega\} \mid x = \omega \vee \exists y : X \bullet x = \mathbb{P}(y)\} && (21) \end{aligned}$$

To see, for example, that ϕ_{5} is not typable, assume it has type τ and consider the singleton set $\{\mathbb{P}(\tau(\mathit{1}))\}$, then we have the contradiction:

$$\mathbb{P}(\tau(\mathit{1})) = \bigcup\{\mathbb{P}(\tau(\mathit{1}))\} \subseteq \tau(\{\mathbb{P}(\tau(\mathit{1}))\}) \simeq \tau(\mathit{1}) \qquad (22)$$

where the final bijection arises because $\{\mathbb{P}(\tau(\mathit{1}))\}$ and $\mathit{1}$ are singleton sets and $X \mapsto \tau(X)$ is a functor which preserves bijections. ϕ_{5} is monotonic and its only fixed point is the empty set if we assume the axiom of foundation.

ϕ_{6} is a finitary operator whose least fixed point on the nose in ZF cannot be shown to exist in Zermelo set theory. To see this observe that $\phi_{\mathit{6}}(X)$ is the union of the singleton set $\{\omega\}$ and the set of power sets of elements of X. The construction of the least fixed point from below produces the countably infinite set $\{\omega, \mathbb{P}(\omega), \mathbb{P}^{\mathit{2}}(\omega), \mathbb{P}^{\mathit{3}}(\omega), \ldots\}$. The union of this is the standard model, $V_{\omega+\omega}$, for Zermelo set theory (see, e.g., [2]). Since Zermelo set theory is closed under unions, it follows that one cannot show in that system that the least fixed point exists. ϕ_{6} does however have fixed points up to a bijection. For example, the countable set, $\mathbb{F}(\omega)$, of finite sets of natural numbers has $\phi(\mathbb{F}(\omega))$ equal to the set of x such that x is the power set of a finite set of natural numbers or $x = \omega$, so that $\phi(\mathbb{F}(\omega))$ is also countable, whence $\mathbb{F}(\omega) \simeq \phi(\mathbb{F}(\omega))$.

Example 7 in section 3 above shows that there are *cts* operators which do not have fixed points in Zermelo even up to a bijection.

7 Fixed Points of Typable Finitary Operators

If the finitary operator ϕ is typable, then, I claim, the existence of a fixed point up to a bijection for ϕ can be proved in Zermelo set theory with choice. We prove this by using the type to construct a set X and an injection from $\phi(X)$ to a subset of X, and then applying lemma 2.

For a given typable finitary ϕ, we construct an X which is ϕ-closed up to a bijection by observing that for big enough X, $\phi(X)$ cannot be very much bigger. To prove this we first give a bound on $\phi(Y)$ for finite Y:

Lemma 3 *Let the monotonic operator ϕ be typable with type τ and let A be finite, then $\phi(A) \simeq Y$ for some subset Y of $\tau(\omega)$.*

Proof: If A has n elements, then we have:

$$\phi(A) \subseteq \tau(A) \simeq \tau(\{\mathit{1}..n\}) \subseteq \tau(\omega) \tag{23}$$

□

Note that in Zermelo set theory it may not be possible to state in full generality propositions like the previous lemma which make assertions about $\tau(X)$ for all types τ (since the axiom of replacement is needed to form the set of all such $\tau(X)$). Thus the lemma should be thought of as the metatheorem: "*For any operator ϕ and for any type τ, it is provable in Zermelo set theory that 'if ϕ is monotonic and typable with type τ, then for any finite A,there exists a subset Y of $\tau(\omega)$ such that $\phi(A) \simeq Y$'*". When such propositions are proved by induction over the structure of τ, we should think of the induction as being carried out in the metalanguage to demonstrate that for any particular τ the relevant proof in Zermelo set theory may be constructed.

As in Z, let us write $\mathbb{F}(X)$ for the set of finite subsets of X. We can now give a bound on $\phi(X)$ for any X, namely that if ϕ has type τ, $\phi(X)$ is no bigger than $\mathbb{F}(X) \times \tau(\omega)$. The proof is just a more careful formulation of the following cardinality argument:

$$\begin{aligned}
\mathsf{card}(\phi(X)) &\leq \mathsf{card}(\bigcup\{\phi(Y) | Y \in \mathbb{F}(X)\}) \\
&\leq \mathsf{card}(\mathbb{F}(X)) \times \mathsf{sup}\{\mathsf{card}(\phi(Y)) | Y \in \mathbb{F}(X)\} \\
&\leq \mathsf{card}(\mathbb{F}(X)) \times \mathsf{card}(\tau(\omega))
\end{aligned}$$

where the first inequality follows from the finitariness of ϕ, the second from elementary set theory, and the third from lemma 3.

Lemma 4 (AC) *Let the finitary operator ϕ be typable with type τ then for any X, $\phi(X) \simeq Y$ for some subset Y of $\mathbb{F}(X) \times \tau(\omega)$.*

Proof: Using lemma 3 and the axiom of choice, choose for each finite subset Y of X an injection $f_Y : \phi(Y) \to \tau(\omega)$. Define $A \subseteq \mathbb{F}(X) \times \tau(\omega)$ by

$$A = \{(Y, n) \in \mathbb{F}(X) \times \tau(\omega) | \exists x \in \phi(Y) \bullet n = f_Y(x)\} \tag{24}$$

and define $g : A \to \phi(X)$ by

$$g(Y, n) = f_Y^{-\mathit{1}}(n) \tag{25}$$

then g is a surjection, since, by finitariness, any $x \in \phi(X)$ is an element of $\phi(Y)$ for some finite Y and then $x = g(Y, f_Y(x))$. Thus we have exhibited a surjection from a subset of $\mathbb{F}(X) \times \tau(\omega)$ onto $\phi(X)$. Using the axiom of choice once more, g has a left inverse which gives the bijection we need. □

Now we can prove the theorem on typable finitary operators:

Theorem 1 (AC) *Let the finitary operator ϕ be typable with type τ, then for some subset X of $\tau(\omega)$, we have $X \simeq \phi(X)$*

Proof: By lemma 4, for some subset, Y say, of $\mathbb{F}(\tau(\omega)) \times \tau(\omega)$ we have:

$$\phi(\tau(\omega)) \simeq Y \subseteq \mathbb{F}(\tau(\omega)) \times \tau(\omega) \simeq \tau(\omega) \times \tau(\omega) \simeq \tau(\omega) \tag{26}$$

where the last two bijections are justified by observing that (i) if a set A is infinite then $\mathbb{F}(A) \simeq A \simeq A \times A$ and (ii) $\tau(\omega)$ is infinite (by an evident induction on the structure of τ).

From (26) we see that there is an injection, say $f : \phi(\tau(\omega)) \to \tau(\omega)$. By lemma 2, the result follows. □

8 Fixed Points of Typable Cts Operators

We now consider typable cts operators. We will show that for each type, τ, there is a finitary operator, ϕ_τ, of type τ, such that for any cts operator, ϕ, of type τ, and any set X, $\phi(X) \subseteq \phi_\tau(X)$. Since finitary operators have fixed points in Zermelo set theory by the results of section 7, it will follow, using lemma 2, that any cts operator has a fixed point too. Indeed the results of section 7 give us a bound on the size of a least fixed point in terms of the type of the operator.

Let us define functions $\mathsf{content}^Y_\tau : \tau(Y) \to \mathbb{P}(Y)$, for each type τ and each set Y, by induction over the structure of τ as follows:

$$\begin{aligned}
\mathsf{content}^Y_X &= x \mapsto \{x\} && (27)\\
\mathsf{content}^Y_\omega &= n \mapsto \varnothing && (28)\\
\mathsf{content}^Y_{(\tau_1 \sqcup \tau_2)} &= \iota_j(x) \mapsto \mathsf{content}^Y_{\tau_j}(x) && (29)\\
\mathsf{content}^Y_{(\tau_1 \times \tau_2)} &= (x_1, x_2) \mapsto \mathsf{content}^Y_{\tau_1}(x_1) \cup \mathsf{content}^Y_{\tau_2}(x_2) && (30)\\
\mathsf{content}^Y_{(\mathbb{P}\tau)} &= s \mapsto \bigcup\{\mathsf{content}^Y_\tau(x) | x \in s\} && (31)
\end{aligned}$$

where the ι_j ($j = 1, 2$) are the injections of the summands into the disjoint union. Note that the value of $\mathsf{content}^Y_\tau(x)$ is independent of the Y we choose (provided we choose it so that $x \in \tau(Y)$), i.e. $\mathsf{content}^Y_\tau(x)$ is monotonic in Y.

Intuitively, the function $\mathsf{content}^Y_\tau$ sends an element x of $\tau(Y)$ to the set of raw materials in Y with which x has been built. For example, we have the following lemma:

Lemma 5 *If $A \in \tau(X)$ and $C = \mathsf{content}^X_\tau(A)$, then $A \in \tau(C)$.*

Proof: The proof is straightforward by induction over the structure of τ. As an example of one of the inductive steps, consider $A \in (\mathbb{P}\tau)(X)$, i.e. $A \in \mathbb{P}(\tau(X))$: by the definition of $\mathsf{content}^X_{(\mathbb{P}\tau)}$,

$$C = \bigcup\{\mathsf{content}^X_\tau(x) | x \in A\}$$

By the inductive hypothesis, we have for each $x \in A$ that $x \in \tau(\mathsf{content}^X_\tau(x))$, so $x \in \tau(C)$, whence $A \subseteq C$, so that $A \in (\mathbb{P}\tau)(C)$ as required. □

Now let us define, for each type τ, an operator ϕ_τ, as follows:

$$\phi_\tau(X) = \{A \in \tau(X) | \mathsf{content}^X_\tau(A) \text{ is finite}\} \tag{32}$$

So, for example:

$$\begin{aligned} \phi_\omega(X) &= \omega \\ \phi_X(X) &= X \\ \phi_{(\mathbb{P}X)}(X) &= \mathbb{F}(X) \end{aligned}$$

Lemma 6 *For any type τ, ϕ_τ is finitary.*

Proof: Let $A \in \phi_\tau(Y)$. We have to exhibit a finite subset, C say, of Y with $A \in \phi_\tau(C)$. Now, by definition of ϕ_τ, $C = \mathsf{content}^Y_\tau(A)$ is a finite subset of Y, and, by lemma 5, $A \in \tau(C)$. Clearly, as C is finite, $\phi_\tau(C) = \tau(C)$ so $A \in \phi_\tau(C)$ as required. □

Lemma 7 *If the cts operator ϕ is typable with type τ, then for any X, $\phi(X) \subseteq \phi_\tau(X)$.*

Proof: We have to show that every $A \in \phi(X)$ has $\mathsf{content}^X_\tau(A)$ finite. Assume for a contradiction that some $A \in \phi(X)$ has $\mathsf{content}^X_\tau(A)$ infinite. Let $\{x_0, x_1, \ldots\}$, where the x_i are pairwise distinct, be a countably infinite subset of A and define a chain, $X_0 \subseteq X_1 \subseteq \ldots$, of subsets of X as follows:

$$\begin{aligned} X_0 &= X \setminus \{x_0, x_1, \ldots\} \\ X_{i+1} &= X \cup \{x_i\} \end{aligned}$$

Then $X = \bigcup_i X_i$, and so, as ϕ is cts, $\phi(X) = \bigcup_i \phi(X_i)$, whence $A \in \phi(X_k) \subseteq \tau(X_k)$ for some k. But this implies that $\mathsf{content}^X_\tau(A) = \mathsf{content}^{X_k}_\tau(A) \in \mathbb{P}(X_k)$ which is impossible since $x_k \in \mathsf{content}^X_\tau(A)$ and $x_k \notin X_k$. □

We may now prove our theorem on cts typable operators. Note that the other results in this section have not used the axiom of choice.

Theorem 2 (AC) *Let the cts operator ϕ be typable with type τ, then for some subset Y of $\tau(\omega)$ we have $Y \simeq \phi(X)$.*

Proof: Using lemma 6 and theorem 1, we find an A, say, such that $A \simeq \phi_\tau(A)$. Using lemma 7, we have that $\phi(A) \subseteq \phi_\tau(A)$, and so, composing this inclusion with a bijection between $\phi_\tau(A)$ and A, we have an injection of $\phi(A)$ into A. The result now follows from lemma 2. □

9 Conclusions

We have shown that fixed points may be constructed in Zermelo set theory for operators which are typable in an appropriate sense and which satisfy a condition corresponding to the criterion for the consistency of free type definitions described in the Z reference manual, [5]. Thus, free type definitions satisfying this criterion are conservative extensions over Z considered as a variant syntax for Zermelo set theory, with the axiom of choice.

There is some interest in giving semantics for Z using as a metalanguage a suitable variant of simple type theory such as Mike Gordon's HOL (see [6]) in such a way that distinct Z types are represented by distinct metalanguage types. The proofs we have given should be adaptable to this context (because all of the constructions used appear to be permissible within a typed framework). The axiom of choice is required in a form which allows the appropriate typed analogues of the cardinality arguments in section 7 to be carried out.

There may be a requirement in Z or similar languages to supply a definitional mechanism like the Z free type definitions which permits the new types to be polymorphic, e.g. to support definitions of a type of labelled trees with the type of the labels as a parameter to the type. I believe the proofs given above can be generalised to this case by extending the definition of a type to allow a finite set of additional, unspecified, ground types corresponding to the type parameters. Care would be needed to generalise the cardinality arguments correctly.

Finally, lemma 7, which was proved without the axiom of choice, may be of some interest in its own right in the development of the Z language. It shows that allowing free type definitions corresponding to operators which are cts but not finitary does not actually increase the expressive power of the language from a theoretical point of view, since the type defined by such an operator is in bijection with a subset of a type which can be defined by a finitary operator. Since all the operators which seem to arise in the practical use of Z are actually finitary, the simpler notion might be the preferred proof obligation to impose in the definition of Z. I understand that this change is already under consideration for reasons of simplicity for the second edition of the Z reference manual.

References

[1] J. Barwise, editor. *Handbook of Mathematical Logic*, volume 90 of *Studies in Logic and the Foundations of Mathematics.* North Holland, 1977.

[2] Kenneth Kunen. *Set Theory: An Introduction to Independence Proofs.* North Holland, 1980.

[3] A. Smith. *On Recursive Free Types in Z.* RSRE Memorandum 91028. MOD PE, RSRE, 1991.

[4] J.M. Spivey. *Understanding Z.* Cambridge University Press, 1988.

[5] J.M. Spivey. *The Z Notation: A Reference Manual.* Prentice-Hall, 1989.

[6] *The HOL System: Description.* SRI International, 4 December 1989.