Collection Principles in Dependent Type Theory*

Peter Aczel[1] and Nicola Gambino[2]

[1] Departments of Mathematics and Computer Science, University of Manchester,
e-mail: petera@cs.man.ac.uk
[2] Department of Computer Science, University of Manchester,
e-mail: ngambino@cs.man.ac.uk

Abstract. We introduce logic-enriched intuitionistic type theories, that extend intuitionistic dependent type theories with primitive judgements to express logic. By adding type theoretic rules that correspond to the collection axiom schemes of the constructive set theory **CZF** we obtain a generalisation of the type theoretic interpretation of **CZF**. Suitable logic-enriched type theories allow also the study of reinterpretations of logic. We end the paper with an application to the double-negation interpretation.

Introduction

In [1] the constructive set theory **CZF** was given an interpretation in the dependent type theory $\mathbf{ML_1V}$. This type theory is a version of Martin-Löf's intuitionistic type theory with one universe of small types, but no W-types except for the special W-type V which is used to interpret the universe of sets of **CZF**. In [2] the interpretation was extended to an interpretation of $\mathbf{CZF^+} = \mathbf{CZF} + \mathbf{REA}$ in $\mathbf{ML_1WV}$. Here **REA** is the Regular Extension Axiom and $\mathbf{ML_1WV}$ is obtained from $\mathbf{ML_1V}$ by adding rules to express that the universe of small types is closed under the formation of W-types, although there are no rules for the general formation of W-types, except for the special W-type V.

In intuitionistic type theories such as $\mathbf{ML_1V}$ logic is usually represented using the propositions-as-types idea. This is indeed how the intuitionistic logic of **CZF** is interpreted in $\mathbf{ML_1V}$. The propositions-as-types interpretation of logic plays an important role in the type theoretic interpretation of **CZF**. Recent work by I. Moerdijk and E. Palmgren shows however that it is possible to interpret **CZF** in predicative categorical universes in which logic is not interpreted using the propositions-as-types idea [17, 18].

One of the aims of this paper is to show how the interpretation of **CZF** can be generalised to an interpretation of **CZF** in a logic-enriched intuitionistic type

* This paper was written while visiting the Mittag-Leffler Institute, The Royal Swedish Academy of Sciences. Both authors wish to express their gratitude for the invitation to visit the Institute. The first author is also grateful to his two departments for supporting his visit. The second author is grateful to his department and to the "Fondazione Ing. A. Gini" for supporting his visit.

theory $\mathbf{ML(CZF)}$, which itself has a natural interpretation back into $\mathbf{CZF}$. By a logic-enriched intuitionistic type theory we mean a pure intuitionistic type theory like $\mathbf{ML_1V}$ that is extended with extra judgement forms to express, relative to a context of variable declarations, being a proposition and assertions that one proposition follows from others.

We expect this generalisation to be fruitful. We will give just one indication of this in the paper. We will show how a logic-enriched type theory can accomodate the j-translation reinterpretations of logic. The idea of a j-translation is really a folklore idea to generalise the double negation translation of classical logic into intuitionistic logic by using any defined modal operator j to generalise the double negation operator provided that j satisfies suitable conditions that correspond to the conditions for a Lawvere-Tierney topology in an elementary topos or for a nucleus on a frame. In general only intuitionistic logic gets reinterpreted. See [5] for a recent treatment.

In order to carry through our interpretation of $\mathbf{CZF}$ in a logic-enriched type theory it will be necessary to include two rules in the type theory corresponding to the two collection schemes of $\mathbf{CZF}$, Strong Collection and Subset Collection. In the original type theoretic interpretation these new type theoretic rules, called Collection rules, were not needed as they are both consequences of the type theoretic axiom of choice that holds in the propositions-as-types interpretation of logic. Fortunately we will see that the type theoretic Collection rules are preserved in a j-translation, although an extra condition on j is required to show that the rule corresponding to Subset Collection is preserved. A set theoretical counterpart of this condition was used in [10] to develop frame-valued semantics for $\mathbf{CZF}$.

We have discussed the issue of translating $\mathbf{CZF}$ into an intuitionistic type theory. It is also natural to consider translations in the opposite direction. For example we may interpret types as sets and objects of a type as elements of the corresponding set. Then to each type forming operation there is the natural set forming operation that corresponds to it. For example corresponding to the Σ and Π forms of type are their set theoretical versions. In this way we get a conceptually very simple set theoretical interpretation of the type theory $\mathbf{ML}$, which has no universes or W-types, in $\mathbf{CZF}$, and this extends to an interpretation of $\mathbf{MLW}$ in $\mathbf{CZF^+}$ and $\mathbf{MLW_1}$ in an extension $\mathbf{CZF^+u}$ of $\mathbf{CZF^+}$ expressing the existence of a universe in the sense of [19]. Here $\mathbf{MLW}$ is $\mathbf{ML}$ with W-types and $\mathbf{MLW_1}$ is its extension with a universe of small types reflecting all the forms of type of $\mathbf{MLW}$. The syntactic details of this kind of translation can be found in [4].

A weakness of these types-as-sets interpretations, when linked with the reverse sets-as-trees interpretations, is that there seems to be a mismatch between the set theories and the type theories. So although we get a translation of $\mathbf{CZF}$ into $\mathbf{ML_1V}$ we only seem to get a translation of $\mathbf{ML}$ into $\mathbf{CZF}$ and to translate $\mathbf{ML_1V}$ into a constructive set theory using types-as-sets we seem to need to go to the set theory $\mathbf{CZF^+u}$, which is much stronger than $\mathbf{CZF}$. This mismatch is overcome in [4] by having axioms for an infinite hierarchy of universes on both the type theory side and the set theory side. This allows for the two sides to catch up with each other.

This brings us to the second main aim of this paper, which is to present another approach to resolving the mismatch problem by replacing the types-as-sets approach by a types-as-classes approach to interpreting type theories in set theories. To carry this approach through it is necessary to restrict the formation of Π-types. This is because in set theory we may only form the class $\Pi_{x \in A} B_x$, where A is a class and B_x is a class for each $x \in A$, when the class A is a set. The corresponding restriction in type theory is to require that $(\Pi x : A)B(x)$ be allowed as a type only when B is a family of types indexed by a small type A. When W-types are also wanted then again it is necessary to put a restriction on their formation. This time the restriction is to only allow the formation of a type $(Wx : A)B(x)$ when each type $B(x)$ is small, although the type A need not be small.

With these restrictions we are led to consider a pure type theory $\mathbf{ML_1^-} + \mathbf{W}^-$ and a logic-enriched extension $\mathbf{ML}(\mathbf{CZF})$ and give a types-as-classes interpretation of $\mathbf{ML}(\mathbf{CZF})$ in $\mathbf{CZF}$. With the reverse sets-as-trees interpretation of $\mathbf{CZF}$ in $\mathbf{ML}(\mathbf{CZF})$ we get a match between a type theory and $\mathbf{CZF}$.

Remark. Due to space constraints, unfortunately this paper does not contain any proofs. Most of the type theoretic rules are also omitted. We hope to present in a future occasion a full version of the paper, including detailed proofs and a complete list of type theoretic rules. A draft of the full version is available from the authors' web pages.

Plan of the paper. Section 1 recalls pure type theories and introduces logic-enriched type theories. Section 2 is devoted to the propositions-as-types interpretation of logic-enriched type theories into pure type theories. Section 3 contains the types-as-classes interpretation of logic-enriched type theories into **CZF**. Section 4 presents the Collection rules and their proposition-as-types and types-as-classes interpretation. Section 5 develops a generalised type theoretic interpretation of **CZF** into $\mathbf{ML}(\mathbf{CZF})$. Section 6 discusses j-translation reinterpretations of logic in logic-enriched type theories.

Acknowledgements. We wish to thank Helmut Schwichtenberg for helpful suggestions. The first author wishes to thank Christopher Nix for pointing out some inaccuracies in a preliminary version of the paper. The second author wishes to thank Steve Awodey, Andrej Bauer, Maria Emilia Maietti, Giovanni Sambin and Alex Simpson for useful discussions.

1 Logic-Enriched Type Theories

1.1 Standard Pure Type Theories

A **standard pure type theory** has the forms of judgement $(\Gamma)\ \mathcal{B}$ where Γ is a context consisting of a list of declarations $x_1 : A_1, \ldots, x_n : A_n$ of distinct variables $x_1, \ldots, x_n$, and $\mathcal{B}$ has one of the forms

$$\begin{aligned} A &: \mathsf{type}\,, \\ A = A' &: \mathsf{type}\,, \\ a &: A\,, \\ a = a' &: A\,. \end{aligned}$$

For the context Γ to be well-formed it is required that

$$\begin{array}{rcl} (\,)\ A_1 & : & \mathsf{type}\,, \\ (x_1 : A_1)\ A_2 & : & \mathsf{type}\,, \\ \ldots\ldots & & \\ (x_1 : A_1, \ldots, x_{n-1} : A_{n-1})\ A_n & : & \mathsf{type}\,. \end{array}$$

The well-formedness of each of these forms of judgement has other presuppositions. Thus, in a well-formed context Γ, the judgement $A = A' : \mathsf{type}$ presupposes that $A : \mathsf{type}$ and $A' : \mathsf{type}$, the judgement $a : A$ presupposes that $A : \mathsf{type}$, and the judgement $a = a' : A$ presupposes that $a : A$ and $a' : A$. In the rest of the paper we will prefer to leave out the empty context whenever possible. So $(\,)\ A_1 : \mathsf{type}$, as above, will be written just $A_1 : \mathsf{type}$.

Any standard type theory will have certain general rules for deriving well-formed judgements, each instance of a rule having the form

$$\frac{J_1 \quad \cdots \quad J_k}{J}$$

where $J_1, \ldots, J_k, J$ are all judgements. In stating a rule of a standard type theory it is very convenient to suppress mention of a context that is common to both the premisses and the conclusion of the rule. For example we will write the reflexivity rule for type equality as just

$$\frac{A : \mathsf{type}}{A = A : \mathsf{type}\,.}$$

But in applying this rule we are allowed to infer $(\Gamma)\ A = A : \mathsf{type}$ from $(\Gamma)\ A : \mathsf{type}$ for any well-formed context Γ.

It will be convenient in stating certain results to add the following additional form of judgement to a standard type theory

$$(\Gamma)\ \ B_1, \ldots, B_m \Rrightarrow B$$

where $(\Gamma)\ B_i : \mathsf{type}$ for $i = 1, \ldots, m$ and $(\Gamma)\ B : \mathsf{type}$ are well-formed judgements. The only rule involving this form of judgement is

$$\frac{(y_1 : B_1, \ldots, y_m : B_m)\ b : B}{B_1, \ldots, B_m \Rrightarrow B\,.}$$

As there are no other rules involving the new judgement form this extension of a standard type theory is conservative.

1.2 Review of Some Pure Type Theories

We will use **ML** to stand for a variant of Martin-Löf's type theory without universes or W-types. We prefer to avoid having any identity types. Also, rather than have finite types $\mathbb{N}_k$ for all $k = 0, 1, \ldots$ we will just have them for $k = 0, 1, 2$ and use the notation $\mathbb{0}, \mathbb{1}, \mathbb{2}$ for them. As usual we define binary product and function types as follows:

$$A_1 \times A_2 \stackrel{\text{def}}{=} (\Sigma \_ : A_1)A_2 \, ,$$
$$A_1 \to A_2 \stackrel{\text{def}}{=} (\Pi \_ : A_1)A_2 \, ,$$

where the symbol $\_$ indicates an anonymous bound variable. Finally we do not take binary sums as primitive but define them. To do so we will allow dependent types to be defined by cases on $\mathbb{2}$; i.e. given $A_1, A_2 : \mathsf{type}$ we allow the formation of $\mathbb{R}_2(A_1, A_2, c) : \mathsf{type}$ whenever $c : \mathbb{2}$ so that for $i = 1, 2$,

$$\mathbb{R}_2(A_1, A_2, i) = A_i : \mathsf{type},$$

where $1, 2 : \mathbb{2}$ are the canonical elements of $\mathbb{2}$. Using $\mathbb{R}_2$ we define, for types A_1, A_2,

$$A_1 + A_2 \stackrel{\text{def}}{=} (\Sigma z : \mathbb{2})\mathbb{R}_2(A_1, A_2, z) \, .$$

So the primitive forms of type of **ML** are

$$\mathbb{0}, \mathbb{1}, \mathbb{2}, \mathbb{N}, \mathbb{R}_2(A_1, A_2, e), (\Sigma x : A)B, (\Pi x : A)B \, .$$

The type theory $\mathbf{ML_1}$ is the pure standard theory obtained from **ML** by adding a type universe $\mathbb{U}$ of small types, or rather of representatives for small types as we will use the universe, *à la* Tarski, where for each $a : \mathbb{U}$, $\mathbb{T}(a)$ is the small type represented by a. The rules for $\mathbb{U}$ express that $\mathbb{U}$ reflects all the forms of type of **ML**.

The type theory **MLW** is obtained from **ML** by adding rules for the W-types $(Wx : A)B$ and $\mathbf{MLW_1}$ is like $\mathbf{ML_1}$ except that the rules for the W-types are added and the type universe $\mathbb{U}$ also reflects the W-types[1]. There are two natural subtheories of $\mathbf{MLW_1}$. The first one[2], $\mathbf{ML_1} + \mathbf{W}$, has W-types but they are not reflected in $\mathbb{U}$. The second one, $\mathbf{ML_{1W}}$, only has small W-types.

The pure type theory $\mathbf{ML_1} + \mathbf{W}$ is a stronger theory than the set theory **CZF**. By replacing the Π and W forms of type by restricted versions we will get a pure type theory $\mathbf{ML_1^-} + \mathbf{W}^-$ which will have a straightforward translation into **CZF** in which types are interpreted as classes of **CZF** and terms are interpreted as sets of **CZF**. This interpretation is to be discussed in sec. 3. In $\mathbf{MLW_1}$ we

[1] In the literature essentially this theory has been written $\mathbf{ML_1W}$. But this seems a bit misleading as it might suggest that the universe does not reflect the W-types.

[2] It would be natural to name this type theory $\mathbf{ML_1W}$, but this is in conflict with the notation of the literature.

can define the following restricted versions of the Π and W forms of type. For $a : \mathbb{U}$ and $(x : \mathbb{T}(a))\ B : \mathsf{type}$ let

$$(\Pi^{-} x : a)B \stackrel{\text{def}}{=} (\Pi x : \mathbb{T}(a))B\,,$$

and for $A : \mathsf{type}$ and $(x : A)\ b : \mathbb{U}$ let

$$(W^{-} x : A)b \stackrel{\text{def}}{=} (W x : A)\mathbb{T}(b)\,.$$

The type theory $\mathbf{ML}_{\mathbf{1}}^{-} + \mathbf{W}^{-}$ does not have the Π and W forms of type but instead has rules for Π^{-} and W^{-} as primitive type forming operators. Note that the type universe $\mathbb{U}$ still reflects the Π^{-} forms of type, but does not reflect the W^{-} forms.

It seems necessary to add extra elimination rules for the type $\mathbb{2}$ and the W^{-} forms of type. In the case of $\mathbb{2}$ this is so as to be able to derive the elimination rules for the defined binary sums, $A_1 + A_2$. In the case of the W^{-} forms of type we will need double recursion on W^{-}-types when we come to define extensional equality on the W^{-}-type $\mathbb{V}$, to be defined in sec. 5. In both cases the Π^{-}-types do not seem to be enough to get what we want, although Π-types are enough. We can no longer define function types $A_1 \to A_2$ for types A_1, A_2. Instead we define $a \to A'$ for $a : \mathbb{U}$ and $A' : \mathsf{type}$ as follows:

$$a \to A' \stackrel{\text{def}}{=} (\Pi^{-} \_ : a)A'\,.$$

1.3 Adding Predicate Logic

Given a standard pure type theory we may consider enriching it with the following two additional forms of judgement, $(\Gamma)\ \mathcal{B}$, where Γ should be a well-formed context as before and $\mathcal{B}$ has one of the following forms:

$$\begin{aligned} &\phi : \ \mathsf{prop}\,, \\ &\phi_1, \ldots, \phi_m \Rightarrow \phi\,. \end{aligned}$$

In the context Γ the well-formedness of $\phi_1, \ldots, \phi_m \Rightarrow \phi$ presupposes that $\phi_i : \mathsf{prop}$ for $i = 1, \ldots, m$ and $\phi : \mathsf{prop}$. Using these new judgement forms it is straightforward to add the standard formation and inference rules for the intuitionistic logical constants; i.e. the canonical true and false propositions $\top, \bot$, the binary connectives $\wedge, \vee, \supset$ and the quantifiers $(\forall x : A), (\exists x : A)$ for each type A. As an example, in Table 1 we give formation and inference rules for disjunction and existential quantification.

As always, in the statement of formation rules we suppress a context that is common to the premisses and conclusion. In the inference rules we will also suppress a list of assumptions appearing on the left hand side of $\Rightarrow$ in the logical premisses and conclusion of each inference rule. Moreover we will write $(\Gamma)\ \phi$ rather than $(\Gamma) \Rightarrow \phi$ and just ϕ rather than the judgement $\Rightarrow \phi$.

Given a standard pure type theory $\mathbf{T}$ let $\mathbf{T} + \mathbf{IL}$ be the theory obtained from $\mathbf{T}$ by enriching it with intuitionistic logic as just described. Each type determines

Table 1. Formation and inference rules for disjunction and existential quantification.

$$\frac{\phi_1 : \mathsf{prop} \qquad \phi_2 : \mathsf{prop}}{\phi_1 \vee \phi_2 : \mathsf{prop}} \qquad\qquad \frac{A : \mathsf{type} \qquad (x : A)\ \phi : \mathsf{prop}}{(\exists x : A)\ \phi : \mathsf{prop}}$$

$$\frac{\phi_1 : \mathsf{prop} \qquad \phi_2 : \mathsf{prop} \qquad \phi_i}{\phi_1 \vee \phi_2}\,(i = 1, 2) \qquad\qquad \frac{\phi_1 \vee \phi_2 \qquad \phi_1 \Rightarrow \theta \qquad \phi_2 \Rightarrow \theta}{\theta}$$

$$\frac{a : A \quad (x : A)\ \phi : \mathsf{prop} \qquad \phi[a/x]}{(\exists x : A)\phi} \qquad\qquad \frac{(\exists x : A)\phi \qquad \theta : \mathsf{prop} \qquad (x : A)\ \phi \Rightarrow \theta}{\theta}$$

the proposition that there is an object of that type; i.e. we can associate with each type expression A of $\mathbf{T}$, the formula $!A \stackrel{\text{def}}{=} (\exists \_ : A)\top$ of $\mathbf{T} + \mathbf{IL}$.

The proof of the direction from left to right in the following result involves simple applications of the inference rules for $\top$ and $\exists$. The result in the other direction is an 'Explicit Definability' result that generalises the Explicit Definability for Numbers result for Heyting Arithmetic. The proof for Heyting Arithmetic, as given in 5.10 of Chapter 3 of [21], carries over here.

Proposition 1. *For any standard pure type theory* $\mathbf{T}$*, if* $\mathbf{T} \vdash (\Gamma)\ B_i : \mathsf{type}$ *for* $i = 1, \ldots, m$*, and* $\mathbf{T} \vdash (\Gamma)\ B : \mathsf{type}$ *then*

$$\mathbf{T} \vdash (\Gamma)\ B_1, \ldots, B_m \Rrightarrow B \quad \textit{iff} \quad \mathbf{T} + \mathbf{IL} \vdash (\Gamma)\ !B_1, \ldots, !B_m \Rightarrow !B\,.$$

1.4 Induction Rules

It is natural to extend a standard logic-enriched type theory with additional non-logical rules to express properties of the various forms of type. For example it is natural to add a rule for mathematical induction to the rules concerning the type of natural numbers and there are similar rules for the other inductive forms of type.

So, for each inductive type $C : \mathsf{type}$ of $\mathbf{MLW} + \mathbf{IL}$, if $(z : C)\,\phi : \mathsf{prop}$ and $e : C$ we have the induction rule

$$\frac{\text{Premisses}}{\phi[e/z]}\,,$$

where the inductive types C and the correspondence between the form of C and the premisses is given in Table 2.

Table 2. The Inductive types of **MLW** and the premisses of their induction rules.

C	Premisses
$\mathbb{0}$	None ,
$\mathbb{1}$	$\phi[0/z]$,
$\mathbb{2}$	$\phi[1/z]\,,\phi[2/z]$,
$\mathbb{N}$	$\phi[0/z]\,,(x:\mathbb{N})\ \phi[x/z] \Rightarrow \phi[\mathsf{succ}(x)/z]$,
$(\Sigma x : A)B$	$(x : A, y : B)\ \phi[\mathsf{pair}(x,y)/z]$
$(Wx : A)B$	$(x : A, u : B \to C)\ (\forall y : B)\phi[\mathsf{app}(u,y)/z] \Rightarrow \phi[\mathsf{sup}(x,u)/z]$.

2 Propositions-as-Types

2.1 Propositions-as-Types Interpretation for ML and MLW

We present the familiar propositions-as-types translation, here abbreviated PaT translation, of $\mathbf{ML}+\mathbf{IL}$ into $\mathbf{ML}$. The PaT translation has no effect on the pure $\mathbf{ML}$ part but, relative to any context, associates with each $\phi :\ \mathsf{prop}$ a type $Pr(\phi) : \mathsf{type}$, so that for each derivation of a judgement $(\Gamma)\ \ \phi_1,\ldots,\phi_m \Rightarrow \phi$ in $\mathbf{ML}+\mathbf{IL}$ there is a derivation in $\mathbf{ML}$ of $(\Gamma)\ \ Pr(\phi_1),\ldots,Pr(\phi_m) \Rrightarrow Pr(\phi)$. The PaT translation is defined as follows:

$$\begin{aligned}
Pr(\bot) &\stackrel{\text{def}}{=} \mathbb{0}\\
Pr(\top) &\stackrel{\text{def}}{=} \mathbb{1}\\
Pr(\phi_1 \wedge \phi_2) &\stackrel{\text{def}}{=} Pr(\phi_1) \times Pr(\phi_2)\\
Pr(\phi_1 \vee \phi_2) &\stackrel{\text{def}}{=} Pr(\phi_1) + Pr(\phi_2)\\
Pr(\phi_1 \supset \phi_2) &\stackrel{\text{def}}{=} Pr(\phi_1) \to Pr(\phi_2)\\
Pr((\forall x : A)\phi_0) &\stackrel{\text{def}}{=} (\Pi x : A)Pr(\phi_0)\\
Pr((\exists x : A)\phi_0) &\stackrel{\text{def}}{=} (\Sigma x : A)Pr(\phi_0)
\end{aligned}$$

We will need the following rule ($\mathbf{PaT}$) to state our main result below about the propositions-as-types interpretation

$$\frac{\phi :\ \mathsf{prop}}{\phi\ \equiv\ !\,Pr(\phi)} \qquad (\mathbf{PaT})$$

where, for $\phi,\psi : \mathsf{prop}$, we define $\phi \equiv \psi \stackrel{\text{def}}{=} (\phi \supset \psi) \wedge (\psi \supset \phi)$. Recall the familiar fact that the type theoretic axiom of choice holds in the propositions-as-types interpretation. We express this version of the axiom of choice as the rule:

$$\frac{A : \mathsf{type} \qquad (x : A)\ B : \mathsf{type} \qquad (x : A, y : B)\ \phi : \mathsf{prop}}{(\forall x : A)(\exists y : B)\phi\ \Rightarrow\ (\exists z : C)(\forall x : A)\phi[\mathsf{app}(z,x)/y]} \qquad (\mathbf{AC})$$

where C is $(\Pi x : A)B$

In order to state our result we assume that $\mathbf{T}$ is any standard pure theory that includes $\mathbf{ML}$. For each raw judgement J of $\mathbf{T}+\mathbf{IL}$ let J^{PaT} be the raw judgement of $\mathbf{T}$ that is just J except when J has either the form $(\Gamma)\ \phi : \mathsf{prop}$ or $(\Gamma)\ \phi_1, \ldots, \phi_m \Rightarrow \phi$ and then J^{PaT} has the form $(\Gamma)\ Pr(\phi) : \mathsf{type}$ or $(\Gamma)\ Pr(\phi_1), \ldots, Pr(\phi_m) \Rrightarrow Pr(\phi)$ respectively. For the next result we need the rule $(\mathbb{0}\perp)$.

$$\frac{a : \mathbb{0}}{\perp} \qquad (\mathbb{0}\perp)$$

Note that, given our abbreviatory conventions, the rule $(\mathbb{0}\perp)$ allows us to infer $(\Gamma)\,\phi_1, \ldots, \phi_n \Rightarrow \perp$ from $(\Gamma)\,a : \mathbb{0}$ and $(\Gamma)\,\phi_i : \mathsf{prop}$ for $i = 1, \ldots, n$.

Theorem 2. *Let* $\mathbf{T}$ *be as above. Then*

1. $\mathbf{T}+\mathbf{IL}+(\mathbf{PaT}) \vdash J$ *implies* $\mathbf{T} \vdash J^{\mathrm{PaT}}$.
2. *In* $\mathbf{T}+\mathbf{IL}$*, the rule* $(\mathbf{PaT})$ *is equivalent to the combination of rules* $(\mathbf{AC})$ *and* $(\mathbb{0}\perp)$.

The main work in proving part 1 is first to show how each logical inference rule translates into a derived rule of $\mathbf{ML}$ following the well-known propositions-as-types idea and second to observe that the instances of $(\mathbf{PaT})$ translate into instances of the following derived rule

$$\frac{A : \mathsf{type}}{A \leftrightarrow \mathbb{1} \times A}$$

where, for types A, B, $A \leftrightarrow B \stackrel{\mathrm{def}}{=} (A \to B) \times (B \to A)$. Provided that $\mathbf{T}$ does not have any additional rules for forming the types of $\mathbf{ML}$ the converse implication to part 1 holds. For part 2, the rule $(\mathbf{PaT})$ can be used to prove $(\mathbf{AC})$ as in Martin-Löf's original proof of the type theoretic axiom of choice in his type theory and the rule $(\mathbb{0}\perp)$ is derived using the instance of $(\mathbf{PaT})$ when ϕ is $\perp$. For the other direction of part 2, $(\mathbf{PaT})$ is proved by induction on the formation of the formula ϕ. The rule $(\mathbb{0}\perp)$ is needed to deal with $\perp$ and $(\mathbf{AC})$ is needed to deal with the implication and universal quantification cases.

Theorem 3. *The induction rules for the inductive types of* $\mathbf{ML}$ *or* $\mathbf{MLW}$ *can be derived in* $\mathbf{ML}+\mathbf{IL}+(\mathbf{PaT})$ *or* $\mathbf{MLW}+\mathbf{IL}+(\mathbf{PaT})$*, respectively.*

This result expresses the familiar observation that each instance of the induction rule for an inductive type comes from an instance of the elimination rule for that type when treating propositions as types.

2.2 A Proposition Universe and its PaT translation

When adding logic to a standard pure type theory $\mathbf{T}$ that includes $\mathbf{ML_1}$ it is natural to also add a proposition universe $\mathbb{P}$ to match the type universe $\mathbb{U}$. The formation rule for this type is

$$\mathbb{P} : \mathsf{type}\,.$$

Elements of this type are to be thought of as representatives for propositions whose quantifiers range over small types. Introduction rules for $\mathbb{P}$ are straightforward. Each object $a : \mathbb{P}$ represents a proposition $\tau(a) :$ prop. The elimination rule for the type $\mathbb{P}$ is the following:

$$\frac{a : \mathbb{P}}{\tau(a) : \mathsf{prop}\,.}$$

For the type $\mathbb{P}$, it seems convenient to avoid the use of an equality form of judgement for propositions in order to express that $\mathbb{P}$ reflects logic. Instead we use logical equivalence. As examples of these rules, in Table 3 we give rules for disjunction and existential quantification.

Table 3. Disjunction and existential quantification in the proposition universe.

$$\frac{p_1 : \mathbb{P} \qquad p_2 : \mathbb{P}}{p_1 \dot{\vee}\, p_2 : \mathbb{P}} \qquad\qquad \frac{a : \mathbb{U} \qquad (x : \tau(a))\ p : \mathbb{P}}{(\dot{\exists} x : a)p : \mathbb{P}}$$

$$\frac{p_1 : \mathbb{P} \qquad p_2 : \mathbb{P}}{\tau(p_1 \dot{\vee}\, p_2) \equiv \tau(p_1) \vee \tau(p_2)} \qquad\qquad \frac{a : \mathbb{U} \qquad (x : \tau(a))\ p : \mathbb{P}}{\tau(\dot{\exists} x : a)p \equiv (\exists x : \tau(a))\tau(p)}$$

When the pure type theory $\mathbf{T}$ includes $\mathbf{ML_1}$ then we write $\mathbf{T} + \mathbf{IL_1}$ for the enrichment of $\mathbf{T}$ with intuitionistic predicate logic and also the rules for $\mathbb{P}$, as illustrated in Table 3. We now wish to give a translation of $\mathbf{T} + \mathbf{IL_1}$ into $\mathbf{T} + \mathbf{IL}$ by interpreting $\mathbb{P}$ as $\mathbb{U}$ following the propositions-as-types idea. Each new symbol of $\mathbf{T} + \mathbf{IL_1}$ that was added to $\mathbf{T} + \mathbf{IL}$ is reinterpreted as a primitive or defined symbol of $\mathbf{T} + \mathbf{IL}$ according to the following correspondence:

$$\begin{array}{ccccccccc} \mathbb{P} & \tau & \dot{\bot} & \dot{\top} & \dot{\wedge} & \dot{\vee} & \dot{\supset} & \dot{\forall} & \dot{\exists} \\ \hline \mathbb{U} & \tau^* & \dot{\mathbb{0}} & \dot{\mathbb{1}} & \dot{\times} & \dot{+} & \dot{\rightarrow} & \dot{\Pi} & \dot{\Sigma} \end{array}$$

where the symbols $\tau^*, \dot{\times}, \dot{+}, \dot{\rightarrow}$ are defined in $\mathbf{T} + \mathbf{IL}$ as follows:

$$\begin{array}{rrl} (x : \mathbb{U}) & \tau^*(x) & \stackrel{\text{def}}{=} !\mathbb{T}(x) : \mathsf{prop}\,, \\ (x_1, x_2 : \mathbb{U}) & x_1 \dot{\times} x_2 & \stackrel{\text{def}}{=} (\dot{\Sigma}\_ : x_1)x_2 : \mathbb{U}\,, \\ (x_1, x_2 : \mathbb{U}) & x_1 \dot{+} x_2 & \stackrel{\text{def}}{=} (\dot{\Sigma} z : \dot{2})\mathbb{R}_2(x_1, x_2, z) : \mathbb{U}\,, \\ (x_1, x_2 : \mathbb{U}) & x_1 \dot{\rightarrow} x_2 & \stackrel{\text{def}}{=} (\dot{\Pi}\_ : x_1)x_2 : \mathbb{U}\,. \end{array}$$

For each expression M of $\mathbf{T} + \mathbf{IL_1}$ let us write M^* for the result of this reinterpretation of the symbols occuring in M. For each raw judgement J of $\mathbf{T} + \mathbf{IL_1}$ let J^* be the raw judgement of $\mathbf{T} + \mathbf{IL}$ obtained by this reinterpretation of the symbols in J.

The PaT translation of $\mathbf{T} + \mathbf{IL}$ into $\mathbf{T}$ extends to a translation of $\mathbf{T} + \mathbf{IL_1}$ into $\mathbf{T}$ if we define

$$Pr(\tau(a)) \stackrel{\mathrm{def}}{=} \mathbb{T}(a^*)$$

for each raw term a of $\mathbf{T} + \mathbf{IL_1}$. To state our next result we need the rule $(\mathbb{P}^*)$.

$$\frac{a : \mathbb{P}}{\tau(a) \equiv \tau^*(a^*)\,.} \qquad (\mathbb{P}^*)$$

Note that the instances of this rule are given by the instances of $(\mathbf{PaT})$ where ϕ has the form $\tau(a)$ for $a : \mathbb{P}$.

Theorem 4. *Let* $\mathbf{T}$ *be any standard pure type theory that includes* $\mathbf{ML_1}$ *and let* J *be any raw judgement of* $\mathbf{T} + \mathbf{IL_1}$*. Then*

1. $\mathbf{T} + \mathbf{IL_1} + (\mathbb{P}^*) \vdash J$ *implies* $\mathbf{T} + \mathbf{IL} \vdash J^*$.
2. $\mathbf{T} + \mathbf{IL_1} + (\mathbf{PaT}) \vdash J$ *implies* $\mathbf{T} + \mathbf{IL} + (\mathbf{PaT}) \vdash J^*$.
3. *In* $\mathbf{T} + \mathbf{IL_1}$ *the rule* $(\mathbf{PaT})$ *is equivalent to the combination of rules* $(\mathbf{AC}) + (\mathbb{P}^*)$.

Combining the first part of theorem 2 with the second part of theorem 4 we get the following result.

Corollary 5. *Let* $\mathbf{T}$ *be a standard pure type theory that includes* $\mathbf{ML_1}$*. Then* $\mathbf{T} + \mathbf{IL_1} + (\mathbf{PaT}) \ \vdash \ J$ *implies* $\mathbf{T} \vdash (J)^{(\mathrm{PaT}_1)}$*, where* $J^{(\mathrm{PaT}_1)}$ *is defined to be the judgement* $(J^*)^{(\mathrm{PaT})}$.

3 Types-as-Classes

In setting up our standard type theories for the purpose of giving a translation into a set theory it will be convenient to have a raw syntax that categorizes each expression into one of the three categories of

- individual expression (i.e. term),
- type expression,
- proposition expression (i.e. formula).

These raw expressions need not be well-formed expressions of the type theory. In fact it is exactly the three judgement forms

- $(\Gamma)\ \ a : A\,,$
- $(\Gamma)\ \ A : \mathsf{type}\,,$
- $(\Gamma)\ \ \phi : \ \mathsf{prop}\,,$

that we use to express that, in the context Γ,

- a is a well-formed term of type A,
- A is a well-formed type,

– ϕ is a well-formed formula.

It will be convenient to call the terms, type expressions and formulae the 0-expressions, 1-expressions and 2-expressions respectively. The raw expressions will be built up from an unlimited supply of individual variables and a signature of constant symbols according to the rules given below. We assume that each constant symbol of the signature has been assigned an arity $(n_1^{\epsilon_1} \cdots n_k^{\epsilon_k})^{\epsilon}$ where $k \geq 0$, $n_1, \ldots, n_k \geq 0$ and each of $\epsilon, \epsilon_1, \ldots, \epsilon_k$ is one of $0, 1, 2$. A symbol of such an arity is **k-place**. The rules for forming raw expressions of the three kinds are as follows.

1. Every variable is a 0-expression.
2. If κ is a constant symbol of arity $(n_1^{\epsilon_1} \cdots n_k^{\epsilon_k})^{\epsilon}$ and, for $i = 1, \ldots, k$, M_i is an ϵ_i-expression and $\vec{x}_i$ is a list of n_i distinct variables then

$$\kappa((\vec{x}_1)M_1, \ldots, (\vec{x}_k)M_k)$$

is an ϵ-expression.

Some conventions. When $k = 0$ then we just write κ rather than $\kappa(\ )$. Also, if some $n_i = 0$ then we write just M_i rather than $(\ )M_i$.

Free and bound occurrences. These are defined in the standard way when the $(\vec{x}_i)$ are treated as variable binding operations, so that free occurrences in M_i of variables from the list $\vec{x}_i$ become bound in $(\vec{x}_i)M_i$ and so also bound in the whole expression $\kappa((\vec{x}_1)M_1, \ldots, (\vec{x}_k)M_k)$.

Substitution. The result $M[M_1, \ldots, M_k/y_1, \ldots, y_k]$ of simultaneously substituting M_i for free occurrences of y_i in M for $i = 1, \ldots, k$, where $y_1, \ldots, y_k$ are distinct variables, is defined in the standard way, relabelling bound variables as usual so as to avoid variable clashes. This is only uniquely specified up to α-convertibility; i.e. up to suitable relabelling of bound variables. In general expressions will be identified up to α-convertibility.

3.1 The Symbols for the Raw Syntax of ML(CZF)

We will eventually be interested in a standard type theory $\mathbf{ML}(\mathbf{CZF})$ which will be obtained from $\mathbf{ML_1^-} + \mathbf{W}^- + \mathbf{IL_1}$ by adding some additional rules including the induction rules for its inductive types, but without adding to its raw syntax.

We now present the symbols of $\mathbf{ML_1^-} + \mathbf{W}^- + \mathbf{IL_1}$, together with their arities. In an arity a missing superscript will be taken to be 0 by default.

0-place symbols
- $0_1, 1, 2, 0, \dot{\mathbb{0}}, \dot{\mathbb{1}}, \dot{\mathbb{2}}, \dot{\mathbb{N}}, \dot{\bot}, \dot{\top}$ of arity $()$,
- $\mathbb{0}, \mathbb{1}, \mathbb{2}, \mathbb{N}, \mathbb{U}, \mathbb{P}$ of arity $()^1$ and $\bot, \top$ of arity $()^2$.

1-place symbols
- succ, R_0 of arity (0), $\mathbb{T}$ of arity $(0)^1$ and τ of arity $(0)^2$.

2-place symbols

- $R_1, \mathsf{pair}, \mathsf{app}, \mathsf{sup}, \dot{\wedge}, \dot{\vee}, \dot{\supset}$ of arity (00) snd $\lambda, \dot{\Sigma}, \dot{\Pi}, \dot{\forall}, \dot{\exists}$ of arity (01),
- split of arity (20) and R_{W^-} of arity (30),
- Σ, Π^-, W^- of arities $(0^1 1^1)^1, (01^1)^1, (0^1 1)^1$ respectively and $\vee, \wedge, \supset$ of arity $(0^2 0^2)^2$,
- $\forall, \exists$ of arity $(0^1 1^2)^2$.

3-place symbols

- $R_2, R_{\mathbb{N}}, \mathbb{R}^+_{W^-}, \mathbb{R}_2$ of arities $(000), (030), (500), (0^1 0^1 0)^1$ respectively.

4-place symbols

- $\mathbb{R}^+_2$ of arity (1100).

Special Conventions. If $\star$ is one of the 2-place symbols $\vee, \wedge, \supset, \dot{\wedge}, \dot{\vee}, \dot{\supset}$ then we use infix notation and write $(M_1 \star M_2)$ rather than $\star(M_1, M_2)$ and if ∇ is any one of $\lambda, \Sigma, \Pi^-, W^-, \forall, \exists, \dot{\Sigma}, \dot{\Pi}, \dot{\forall}, \dot{\exists}$ then we use quantifier notation and write $(\nabla x : M)M'$ rather than $\nabla(M, (x)M')$

3.2 The Set Theoretical Interpretation of Raw Syntax

In the rest of sec. 3 we will work informally in **CZF**. By a **set theoretical sentence** we mean a sentence in the language of **CZF** that may have sets as parameters. By a **variable assignment**, ξ, we mean an assignment of a set $\xi(x)$ to each variable x. The following terminology will be useful. We define a 0-class to be a set, a 1-class to be a class and a 2-class to be a set theoretical sentence. Also, for $n \geq 0$ and $\epsilon = 0, 1, 2$, an n^ϵ-class is a definable operator F assigning an ϵ-class $F(a_1, \ldots, a_n)$ to each n-tuple $(a_1, \ldots, a_n)$ of sets.

Given a signature as above for determining a raw syntax for a type theory we will want to give a set theoretical denotation $[[M]]_\xi$ to each expression M and each variable assignment ξ, so that for each term a its denotation, $[[a]]_\xi$, should be a set, for each type expression A its denotation, $[[A]]_\xi$, should be a class and, for each proposition expression ϕ, its denotation $[[\phi]]_\xi$, should be a set theoretical sentence. We will use structural induction on the way expressions are built up. To do this we will need to have a set theoretical interpretation $\mathcal{F}_\kappa$ for each symbol κ of the signature. Each $\mathcal{F}_\kappa$ has to be a suitable operator so that the second clause of the following definition by structural induction makes sense.

1. $[[M]]_\xi = \xi(x)$ if M is a variable x.
2. $[[M]]_\xi = \mathcal{F}_\kappa(F_1, \ldots, F_k)$ if M is the expression $\kappa((\vec{x}_1)M_1, \ldots, (\vec{x}_k)M_k)$ where κ is a constant symbol of the signature of arity $(n_1^{\epsilon_1} \cdots n_k^{\epsilon_k})^\epsilon$ and, for $i = 1, \ldots, k$, F_i is the $n_i^{\epsilon_i}$-class such that $F_i(\vec{a}_i) = [[M_i]]_{\xi(\vec{a}_i/\vec{x}_i)}$ for all n_i-tuples $\vec{a}_i$ of sets.

When κ has arity $(n_1^{\epsilon_1} \cdots n_k^{\epsilon_k})^\epsilon$ we will require that $\mathcal{F}_\kappa$ is a set operator of that arity. This means that whenever F_i is an $n_i^{\epsilon_i}$-class, for $i = 1, \ldots, k$ then $\mathcal{F}_\kappa(F_1, \ldots, F_k)$ should be an ϵ-class obtained 'uniformly' from $F_1, \ldots, F_k$.

3.3 Soundness

Given a set theoretical interpretation as above of the symbols of a signature that determines a set theoretical denotation $[[M]]_\xi$ to each expression M relative to a variable assignment ξ we can define the following semantic notions.

Definition 6. – *If Γ is $x_1 : A_1, \ldots, x_n : A_n$ then let* $\xi \models \Gamma$ *iff* $\xi(x_i) \in [[A_i]]_\xi$ *for* $i = 1, \ldots, n$ *.*
- *Let* $\xi \models A : \mathsf{type}$ *for any type expression* A*.*
- *Let* $\xi \models A = A' : \mathsf{type}$ *iff* $[[A]]_\xi = [[A']]_\xi$*.*
- *Let* $\xi \models a : A : \mathsf{type}$ *iff* $[[a]]_\xi \in [[A]]_\xi$*.*
- *Let* $\xi \models a = a' : A : \mathsf{type}$ *iff* $[[a]]_\xi = [[a']]_\xi \in [[A]]_\xi$*.*
- *Let* $\xi \models \phi :$ prop *for any formula* ϕ*.*
- *Let* $\xi \models \phi_1 \ldots, \phi_m \Rightarrow \phi$ *if* $[[\phi_1]]_\xi \wedge \cdots \wedge [[\phi_m]]_\xi \supset [[\phi]]_\xi$ *is a set theoretical sentence that is true (in* **CZF***).*

Definition 7. *The raw judgement* $(\Gamma)\mathcal{B}$ *is* **valid** *if* $\xi \models \Gamma$ *implies* $\xi \models \mathcal{B}$ *for every variable assignment* ξ*. A type theory rule is* **sound** *if whenever the premisses of an instance of the rule are valid then so is the conclusion.*

Along the lines of section 2.4 of [4] we can get the following result.

Theorem 8 (CZF). *There is an interpretation of the raw syntax of the type theory* $\mathbf{ML_1^-} + \mathbf{W}^- + \mathbf{IL_1}$ *in* **CZF** *so that each rule of inference of the type theory* **T** *is sound, where* **T** *is obtained from* $\mathbf{ML_1^-} + \mathbf{W}^- + \mathbf{IL_1}$ *by adding the induction rules for its inductive types.*

The interpretation given by this theorem can be rephrased as a syntactic translation into **CZF**.

Corollary 9. *There is a syntactic translation that assigns a sentence "J is valid" of* **CZF** *to each raw judgement* J *of the theory* **T** *of the theorem such that* $\mathbf{T} \vdash J$ *implies* $\mathbf{CZF} \vdash$ *"J is valid".*

4 Collection Principles

The original type theoretic interpretation of **CZF** in $\mathbf{ML_1V}$ rests on two main components. The first component is the definition of a type V, called the type of constructive iterative sets, that is used to interpret the universe of sets of **CZF**. The second component is the propositions-as-types interpretation of logic. This interpretation of logic plays a role in proving the validity of the Restricted Separation, Strong Collection and Subset Collection axiom schemes of **CZF**. Validity of the Restricted Separation axiom scheme follows from the correspondence between restricted propositions and small types. Validity of the Strong Collection and Subset Collection axiom schemes follows instead from the type theoretic axiom of choice, that holds in the propositions-as-types interpretation of logic.

In the following we will present a type theoretic interpretation of **CZF** in a logic-enriched type theory that generalises the original type theoretic interpretation. The generalisation involves treating logic as primitive and not via the propositions-as-types interpretation. In order to do so, we will introduce a logic-enriched type theory called **ML(CZF)**. The type theory **ML(CZF)** extends the logic-enriched type theory **T**, of Theorem 8, with two collection rules, corresponding to the collection axiom schemes of **CZF**. Within the type theory **ML(CZF)** we define a type $\mathbb{V}$, called the type of iterative small classes, that will be used to interpret the universe of sets of **CZF**. The definition of $\mathbb{V}$ allows us to prove the validity of the Restricted Separation axiom scheme without assuming the propositions-as-types interpretation of logic. The collection rules of **ML(CZF)** allow us to prove the validity of the Strong Collection and Subset Collection axiom schemes of **CZF** without assuming the type theoretic axiom of choice.

4.1 The Type of Subsets of a Type

Let us now introduce the type of subsets of a type and define some operations on this type that will be useful in the following. For $A : \mathsf{type}$ we define the **type of subsets** of A, $\mathsf{Sub}(A)$, as follows:

$$\mathsf{Sub}(A) \overset{\text{def}}{=} (\Sigma x : \mathbb{U}) \big((x \to \mathbb{P}) \times (x \to A)\big) .$$

For $a : \mathsf{Sub}(A)$ we define

$$\begin{aligned} \dot{\mathsf{el}}(a) &\overset{\text{def}}{=} a.1 : \mathbb{U} \, , \\ \mathsf{el}(a) &\overset{\text{def}}{=} \mathbb{T}(\dot{\mathsf{el}}(a)) : \mathsf{type} \, , \end{aligned}$$

and for $x : \mathsf{el}(a)$ we define

$$\begin{aligned} \dot{\mathsf{dom}}(a,x) &\overset{\text{def}}{=} \mathsf{app}(a.2.1, x) : \mathbb{P} \, , \\ \mathsf{dom}(a,x) &\overset{\text{def}}{=} \tau(\dot{\mathsf{dom}}(a,x)) : \mathsf{prop} \, , \\ \mathsf{val}(a,x) &\overset{\text{def}}{=} \mathsf{app}(a.2.2, x) : A \, . \end{aligned}$$

Using these definitions, we can informally think of $a : \mathsf{Sub}(A)$ as the 'set' of all objects $\mathsf{val}(a,x) : A$ with $x : \mathsf{el}(a)$ such that $\mathsf{dom}(a,x)$. If $(x : A)\ p : \mathbb{P}$ we define

$$\begin{aligned} (\dot{\forall} x \in a)\ p &\overset{\text{def}}{=} (\dot{\forall} x : \dot{\mathsf{el}}(a))\ \dot{\mathsf{dom}}(a,x) \mathbin{\dot{\supset}} p[\mathsf{val}(a,x)/x] : \mathbb{P} \, , \\ (\dot{\exists} x \in a)\ p &\overset{\text{def}}{=} (\dot{\exists} x : \dot{\mathsf{el}}(a))\ \dot{\mathsf{dom}}(a,x) \mathbin{\dot{\wedge}} p[\mathsf{val}(a,x)/x] : \mathbb{P} \, . \end{aligned}$$

If $(x : A)\ \phi : \mathsf{prop}$ we define

$$\begin{aligned} (\forall x \in a)\ \phi &\overset{\text{def}}{=} (\forall x : \mathsf{el}(a))\ \mathsf{dom}(a,x) \ \supset \ \phi[\mathsf{val}(a,x)/x] : \mathsf{prop} \, , \\ (\exists x \in a)\ \phi &\overset{\text{def}}{=} (\exists x : \mathsf{el}(a))\ \mathsf{dom}(a,x) \ \wedge \ \phi[\mathsf{val}(a,x)/x] : \mathsf{prop} \, . \end{aligned}$$

4.2 The Collection Rules of ML(CZF)

Type theoretic rules corresponding to the collection axiom schemes of **CZF** will now be introduced. The Strong Collection rule corresponds to the Strong Collection axiom scheme and the Subset Collection rule corresponds to the Subset Collection axiom scheme. We will refer to these two rules as Collection rules.

In order to present these rules as simply as possible, let us introduce some definitions. For $A, B : \mathsf{type}$, $a : \mathsf{Sub}(A), b : \mathsf{Sub}(B)$ and $(x : A, y : B)\, \phi : \mathsf{prop}$ we define:

$$\mathsf{coll}(a, b, (x, y)\phi) \stackrel{\text{def}}{=} (\forall x \in a)\, (\exists y \in b)\, \phi \;\wedge\; (\forall y \in b)\, (\exists x \in a)\, \phi \quad : \mathsf{prop}\,.$$

Strong Collection Rule.

$$\frac{A, B : \mathsf{type} \qquad a : \mathsf{Sub}(A) \qquad (x : A, y : B)\, \phi : \mathsf{prop}}{(\forall x \in a)\, (\exists y : B)\, \phi \Rightarrow (\exists v : \mathsf{Sub}(B))\, \mathsf{coll}(a, v, (x, y)\phi)}$$

Subset Collection Rule.

$$\frac{A, B, C : \mathsf{type} \quad a : \mathsf{Sub}(A) \quad b : \mathsf{Sub}(B) \quad (x : A, y : B, z : C)\, \psi : \mathsf{prop}}{(\exists u : \mathsf{Sub}(\mathsf{Sub}(B)))(\forall z : C)\big((\forall x \in a)\, (\exists y \in b)\psi \supset (\exists v \in u)\mathsf{coll}(a, v, (x, y)\psi)\big)}$$

We define the type theory $\mathbf{ML}(\mathbf{CZF})$ as the extension of the type theory $\mathbf{ML}_{\mathbf{1}}^{-} + \mathbf{W}^{-} + \mathbf{IL}_{\mathbf{1}}$, obtained by adding the induction rules for its inductive types and the Strong Collection and the Subset Collection rules. Recall that $\mathbf{CZF}^{-}$ is the subsystem of **CZF** obtained from **CZF** by leaving out the Subset Collection axiom scheme [3, 6]. We define $\mathbf{ML}(\mathbf{CZF}^{-})$ as the type theory obtained from $\mathbf{ML}(\mathbf{CZF})$ by leaving out the Subset Collection rule.

4.3 The Types-as-Classes Interpretation of the Collection Rules

We now work informally in **CZF**. To interpret the operation Sub on types we introduce the corresponding operation Sub on classes, where for each class A we let

$$Sub(A) \stackrel{\text{def}}{=} \Sigma_{I \in V}(Pow(1)^I \times A^I)\ .$$

If $b = (I, (f, g)) \in Sub(A)$ then let

$$set(b) \stackrel{\text{def}}{=} \{g(i) \mid i \in I \;\wedge\; 0 \in f(i)\} \in Sub(A)\ .$$

Conversely, if $a \in Pow(A)$ then let

$$sub(a) \stackrel{\text{def}}{=} (a, ((\lambda\_ \in a)1, (\lambda x \in a)x)) \in Sub(A)\ .$$

Then $set(sub(a)) = a$ for all $a \in Pow(A)$. Recall that, for sets a, b and any set theoretical formula ψ in the two variables x, y we define

$$coll(a, b, (x, y)\psi) \stackrel{\text{def}}{=} (\forall x \in a)(\exists y \in b)\psi \;\wedge\; (\forall y \in b)(\exists x \in a)\psi\ .$$

Theorem 10. *There is an interpretation of the raw syntax of* **ML**(**CZF**) *in* **CZF** *so that each rule of inference is sound.*

Corollary 11. *There is a syntactic translation that assigns a sentence "J is valid" of* **CZF** *to each raw judgement J of* **ML**(**CZF**) *such that* $\mathbf{ML}(\mathbf{CZF}) \vdash J$ *implies* $\mathbf{CZF} \vdash$ *" J is valid".*

4.4 Propositions-as-Types Interpretation of the Collection Rules

The (**AC**) rule cannot be fully formulated when we only have the restricted Π types, as in **ML**(**CZF**). So we need the weakening ($\mathbf{AC}^-$) in order to state the next result.

$$\frac{a : \mathbb{U} \qquad (x : \mathbb{T}(a))\ B : \mathsf{type} \qquad (x : \mathbb{T}(a), y : B)\ \phi :\ \mathsf{prop}}{(\forall x : \mathbb{T}(a))(\exists y : B)\phi\ \Rightarrow\ (\exists z : C)(\forall x : \mathbb{T}(a))\phi[\mathsf{app}(z,x)/y]} \qquad (\mathbf{AC}^-)$$

where C is $(\Pi^- x : a)B$.

Theorem 12. *Let* **T** *be any standard pure type theory that includes* $\mathbf{ML_1^-}$. *Then the Strong Collection and Subset Collection rules are derived rules of* $\mathbf{T} + \mathbf{IL_1} + (\mathbf{AC}^-) + (\mathbb{P}^*)$ *and so they have a* PaT_1 *translation into any standard pure type theory that includes both* **T** *and* $\mathbf{ML_1}$, *where the* PaT_1 *translation was defined in Corollary 5.*

Corollary 13. *The type theory* **ML**(**CZF**) *has a* PaT_1 *translation into the type theory* $\mathbf{ML_1} + \mathbf{W}^-$.

5 A Generalised Type Theoretic Interpretation

In this section we work informally within the type theory **ML**(**CZF**). Our aim is to define an interpretation of **CZF**. In order to do so, we define the type $\mathbb{V}$ of **iterative small classes** as follows:

$$\mathbb{V} \stackrel{\text{def}}{=} \big(W^- y : (\Sigma x : \mathbb{U})(x \to \mathbb{P})\big)\ y.1\,.$$

A canonical iterative small class consists of a small type, a small predicate on the type and a function from the small type to $\mathbb{V}$. A canonical iterative small class $\mathsf{sup}(\mathsf{pair}(a,p),f)$ can be thought of as the 'set' of all $f(x) : \mathbb{V}$ with $x : \mathbb{T}(a)$ such that $p(x)$. By recursion on $\mathbb{V}$ and on $\mathsf{Sub}(\mathbb{V})$ we can define $(x : \mathsf{Sub}(\mathbb{V}))\ \mathsf{set}(x) : \mathbb{V}$ and $(y : \mathbb{V})\ \mathsf{sub}(y) : \mathsf{Sub}(\mathbb{V})$ such that, for $a : \mathbb{U}, b : a \to \mathbb{P}, c : a \to \mathbb{V}$

$$\mathsf{set}(\mathsf{pair}(a, \mathsf{pair}(b,c))) = \mathsf{sup}(\mathsf{pair}(a,b),c) : \mathbb{V}\,,$$
$$\mathsf{sub}(\mathsf{sup}(\mathsf{pair}(a,b),c) = \mathsf{pair}(a, \mathsf{pair}(b,c)) : \mathsf{Sub}(\mathbb{V})\,.$$

Given these definitions, the introduction rule for the type $\mathbb{V}$ can be derived from the following rule:

$$\frac{d : \mathsf{Sub}(\mathbb{V})}{\mathsf{set}(d) : \mathbb{V}}$$

For $(x : \mathbb{V})\ \phi : \mathsf{prop}$ and $(y : \mathsf{Sub}(\mathbb{V}))\ \psi : \mathsf{prop}$ the judgements

$$(\nabla x : \mathbb{V})\, \phi \equiv (\nabla y : \mathsf{Sub}(\mathbb{V}))\, \phi[\mathsf{set}(y)/x]\,,$$

where ∇ is either $\forall$ or $\exists$, are derivable. For $x : \mathbb{V}$, $(y : \mathbb{V})\, p : \mathbb{P}$, and $(y : \mathbb{V})\, \phi : \mathsf{prop}$ we define

$$(\nabla y \in x)\, p \stackrel{\text{def}}{=} (\dot{\forall} y \in \mathsf{sub}(x)) p : \mathbb{P}$$
$$(\nabla y \in x)\, \phi \stackrel{\text{def}}{=} (\forall y \in \mathsf{sub}(x)) \phi : \mathsf{prop}$$

where ∇ is $\forall$ or $\exists$. By double recursion on $\mathbb{V}$ it is possible to define $(x, y : \mathbb{V})\ x \dot{\approx} y : \mathbb{P}$ such that if we let $x \approx y \stackrel{\text{def}}{=} \tau(x \dot{\approx} y)$ then the judgement

$$x \approx x' \equiv \forall y \in x\, \exists y' \in x'\, (y \approx y') \wedge \forall y' \in x'\, \exists y \in x\, (y \approx y')$$

is derivable. We now define the generalized type theoretic interpretation of **CZF**. We assume that **CZF** is formulated in a language with equality, with primitive restricted quantifiers but no membership relation. Membership can easily be defined using equality and existential quantification. In the following, we assume that the symbols for variables for sets of **CZF** coincide with the symbols for variables of type $\mathbb{V}$. We now define two interpretations. A first interpretation, indicated with $[\![\cdot]\!]$, applies to arbitrary formulas, and another interpretation, indicated with $(\!|\cdot|\!)$, applies only to restricted formulas. Both interpretations are defined in table 4, where $\star$ is $\wedge$, $\vee$ or $\supset$, and ∇ is $\forall$ or $\exists$.

Table 4. Interpretation of the language of **CZF**.

$$\begin{aligned}
[\![x = y]\!] &\stackrel{\text{def}}{=} x \approx y & \qquad (\!|x = y|\!) &\stackrel{\text{def}}{=} x \dot{\approx} y\,,\\
[\![\phi_1 \star \phi_2]\!] &\stackrel{\text{def}}{=} [\![\phi_1]\!] \star [\![\phi_2]\!]\,, & (\!|\phi_1 \star \phi_2|\!) &\stackrel{\text{def}}{=} (\!|\phi_1|\!) \dot{\star} (\!|\phi_2|\!)\,,\\
[\![(\nabla x \in y)\, \phi_0]\!] &\stackrel{\text{def}}{=} (\nabla x \in y)\ [\![\phi_0]\!]\,, & (\!|(\nabla x \in y)\ \phi_0|\!) &\stackrel{\text{def}}{=} (\dot{\nabla} x \in y)\ (\!|\phi_0|\!)\,,\\
[\![(\nabla x)\, \phi_0]\!] &\stackrel{\text{def}}{=} (\nabla x : \mathbb{V})\ [\![\phi_0]\!], & &
\end{aligned}$$

Lemma 14. *If ϕ_1 and ϕ_2 are formulas of the language of set theory with free variables $\vec{x}$, and ϕ_2 is restricted, then the judgements*

$$\begin{aligned}
&(\vec{x} : \mathbb{V})\ [\![\, \phi_1 \,]\!] : \mathsf{prop}\,,\\
&(\vec{x} : \mathbb{V})\ (\!|\, \phi_2 \,|\!) : \mathbb{P}\,,\\
&(\vec{x} : \mathbb{V})\ \tau\big((\!|\, \phi_2 \,|\!)\big) \equiv [\![\, \phi_2 \,]\!]
\end{aligned}$$

are derivable.

A formula ϕ with free variables $\vec{x}$ will be said to be **valid** if the judgement

$$(\vec{x} : \mathbb{V}) \quad [\![\phi]\!]$$

is derivable. We say that the generalized type theoretic interpretation of **CZF** is sound if each axiom and each instance of each axiom scheme of **CZF** is valid.

Theorem 15 (ML(CZF)). *The generalized type theoretic interpretation of the set theory* **CZF** *is sound.*

Corollary 16. **CZF** *and* **ML(CZF)** *are mutually interpretable.*

6 Reinterpreting Logic

We now describe how both the logic-enriched type theories **ML(CZF$^-$)** and **ML(CZF)** can accomodate reinterpretations of the logic. We focus our attention on reinterpretations of the logic as determined by an operator j on the type $\mathbb{P}$ that satisfies a type theoretic version of the properties of a Lawvere-Tierney topology in an elementary topos [16] or of a nucleus on a frame [14]. We will call such an operator j a **topology**. The reinterpretation of logic determined by j will be called the j-interpretation.

In discussing j-interpretations, it seems appropriate to consider **ML(CZF$^-$)** initially, and **ML(CZF)** at a later stage. There are two main reasons for doing so. A first reason is that the Strong Collection rule is sufficient to prove the basic properties of j-interpretations. A second reason is that the Strong Collection rule is preserved by the j-interpretation determined by any topology j, while the Subset Collection rule does not seem to be. In order to obtain the derivability of the j-interpretation of the Subset Collection rule, we will introduce a further assumption.

6.1 Topologies in ML(CZF$^-$)

To introduce topologies, for $a, b : \mathbb{P}$ we define $a \leq b \overset{\text{def}}{=} \tau(a) \supset \tau(b) : \mathsf{prop}$.

Definition 17. *Let j be an explicitly defined operator on $\mathbb{P}$, i.e. there is an explicit definition of the form $jx \overset{\text{def}}{=} e : \mathbb{P}$, for $x : \mathbb{P}$, where $(x : \mathbb{P})\, e : \mathbb{P}$. We say that j is a* **topology** *if the following hold for $a_1, a_2 : \mathbb{P}$:*

1. $a_1 \leq ja_1$,
2. $a_1 \leq a_2 \Rightarrow ja_1 \leq ja_2$,
3. $ja_1 \dot\wedge ja_2 \leq j(a_1 \dot\wedge a_2)$,
4. $j(ja_1) \leq ja_1$.

¿From now on we assume given an arbitrary topology j. For $\phi : \mathsf{prop}$, we define

$$J\phi \overset{\text{def}}{=} \exists x : \mathbb{P}\big(\tau(jx) \wedge \tau(x) \supset \phi\big) .$$

Proposition 18. *For* $a : \mathbb{P}$, $J(\tau(a)) \equiv \tau(ja)$.

The properties of j can be lifted to J. It is worth pointing out that the Strong Collection rule is used to prove the fourth part of proposition 19.

Proposition 19. *For* $\phi_1, \phi_2 : \mathsf{prop}$*, the following hold:*

1. $\phi_1 \supset J\phi_1$,
2. $\phi_1 \supset \phi_2 \Rightarrow J\phi_1 \supset J\phi_2$,
3. $J\phi_1 \wedge J\phi_2 \supset J(\phi_1 \wedge \phi_2)$,
4. $J(J\phi_1) \supset J\phi_1$.

We now define the j-interpretation of of $\mathbf{ML}(\mathbf{CZF}^-)$ into itself determined by the topology j. This interpretation acts solely on the logic, leaving types unchanged. We define the j-interpretation $\langle \cdot \rangle_j$ by structural induction on the raw syntax of the type theory. First of all type expressions are left unchanged. Table 5 contains the definition of the interpretation of formulae, where $\star$ is either $\wedge, \vee$ or $\supset$ and ∇ is either $\forall$ or $\exists$, and of judgement bodies.

Table 5. Definition of the j-interpretation of formulae and judgement bodies.

$$\begin{aligned} \langle \top \rangle_j &\stackrel{\text{def}}{=} \top , \\ \langle \bot \rangle_j &\stackrel{\text{def}}{=} \bot , \\ \langle \phi_1 \star \phi_2 \rangle_j &\stackrel{\text{def}}{=} J\langle \phi_1 \rangle_j \star J\langle \phi_2 \rangle_j , \\ \langle (\nabla x : A)\, \phi_0 \rangle_j &\stackrel{\text{def}}{=} (\nabla x : A)\, J\langle \phi_0 \rangle_j , \\ \langle \tau(a) \rangle_j &\stackrel{\text{def}}{=} \tau(a) . \end{aligned}$$

$$\begin{aligned} \langle A : \mathsf{type} \rangle_j &\stackrel{\text{def}}{=} A : \mathsf{type} , \\ \langle A = A' : \mathsf{type} \rangle_j &\stackrel{\text{def}}{=} A = A' : \mathsf{type} , \\ \langle a : A \rangle_j &\stackrel{\text{def}}{=} a : A , \\ \langle a = a' : A \rangle_j &\stackrel{\text{def}}{=} a = a' : A , \\ \langle \phi : \mathsf{prop} \rangle_j &\stackrel{\text{def}}{=} \langle \phi \rangle_j : \mathsf{prop} , \\ \langle \phi_1, \ldots, \phi_n \Rightarrow \phi \rangle_j &\stackrel{\text{def}}{=} J\langle \phi_1 \rangle_j, \ldots, J\langle \phi_n \rangle_j \Rightarrow J\langle \phi \rangle_j . \end{aligned}$$

Finally, we define the j-interpretation of judgements as follows.

$$\langle\ (\Gamma)\ \mathcal{B}\ \rangle_j \stackrel{\text{def}}{=} (\ \Gamma\ )\ \langle \mathcal{B} \rangle_j .$$

Definition 20. *The j-interpretation of a rule*

$$\frac{(\Gamma_1)\ \mathcal{B}_1 \qquad \cdots \qquad (\Gamma_n)\ \mathcal{B}_n}{(\Gamma)\ \mathcal{B}}$$

is said to be **sound** *if the judgement $\langle (\Gamma)\ \mathcal{B} \rangle_j$ is derivable from the judgements $\langle (\Gamma_1)\ \mathcal{B}_1 \rangle_j, \ldots, \langle (\Gamma_n)\ \mathcal{B}_n \rangle_j$.*

It is worth pointing out that the Strong Collection rule implies that its j-interpretation is sound. However, it does not seem possible to prove that the j-interpretation of the Subset Collection rule is sound for an arbitrary topology j. We therefore introduce the following definition.

Definition 21. *A topology j on $\mathbb{P}$ is said to be* **set presented** *by $R : \mathsf{Sub}(\mathbb{P})$ if the judgement*

$$(\forall p : \mathbb{P})\ \big(\ \tau(jp) \equiv (\exists q \in R)\, q \leq p\ \big)$$

is derivable.

The notion of set presented topology is closely related to the notions of cover algebra with basic covers [11], set presented meet semilattice and frame [3, 10] and inductively generated formal topology [8, 20].

Theorem 22 ($\mathbf{ML}(\mathbf{CZF}^{-})$). *Let j be a topology.*

1. *The j-interpretation of each rule of $\mathbf{ML}(\mathbf{CZF}^{-})$ is sound.*
2. *Assuming the Subset Collection rule of $\mathbf{ML}(\mathbf{CZF})$, if j is set presented, then the j-interpretation of the Subset Collection rule is sound.*

6.2 Double Negation Interpretation

As an application of the results just described we present a type theoretic version of the double negation interpretation. We define the double negation topology as follows:

$$(x : \mathbb{P})\ jx \stackrel{\text{def}}{=} \dot{\neg}\dot{\neg}\, x : \mathbb{P}\ ,$$

where $\dot{\neg} x \stackrel{\text{def}}{=} x \mathbin{\dot{\supset}} \dot{\bot} : \mathbb{P}$, for $x : \mathbb{P}$. It is easy to prove that j is a topology. Let us point out that the operator J determined by the double negation topology need not to be logically equivalent to double negation. In fact, for $\phi : \mathsf{prop}$ it holds

$$J\phi \equiv (\exists p : \mathbb{P})\big(\neg\neg\tau(p) \wedge \tau(p) \supset \phi\big)\ ,$$

where $\neg\phi \stackrel{\text{def}}{=} \phi \supset \bot$, for $\phi : \mathsf{prop}$. In general it will hold only that $J\phi$ implies $\neg\neg\phi$ but not viceversa. This fact seems to be one of the reasons for which it is possible to prove the soundness of the j-interpretation of the Strong Collection rule. These observations first arose in connection with the development of frame-valued semantics for **CZF** [10]. The double-negation nucleus on the frame of truth values corresponds closely to a double-negation interpretation [9, 12].

Since the j-interpretation acts as the double negation only on small propositions, it is natural to consider the following principle of **restricted excluded middle**.

$$(x : \mathbb{P})\ \tau(x)\ \vee\ \neg\tau(x) \qquad \textbf{(REM)}$$

Theorem 23. *The $\neg\neg$-interpretation of $\mathbf{ML}(\mathbf{CZF}^{-}) + (\mathbf{REM})$ in $\mathbf{ML}(\mathbf{CZF}^{-})$ is sound.*

Let us now point out that the type theory $\mathbf{ML}(\mathbf{CZF}^-)$ and the theory $\mathbf{CZF}^-$ are mutually interpretable. It is easy to see that $(\mathbf{REM})$ allows to prove in type theory the validity of the principle of excluded middle for restricted formulas. Theorem 23 gives then as a corollary an interpretation of the set theory $\mathbf{CZF}^- + \mathbf{REM}$, obtained from $\mathbf{CZF}^-$ by adding the law of excluded middle for restricted formulae, into $\mathbf{CZF}^-$, a result originally obtained in [7].

We can now consider a type theoretic principle asserting that the double negation topology is set presented:

$$\exists R : \mathsf{Sub}(\mathbb{P})\ \forall p : \mathbb{P}\big(\ \neg\neg\tau(p) \equiv \exists q \in R\,(q \supset \tau(p))\big)\,. \qquad (\mathbf{DNSP})$$

Theorem 24. *The* $\neg\neg$*-interpretation of* $\mathbf{ML}(\mathbf{CZF}) + (\mathbf{REM})$ *in* $\mathbf{ML}(\mathbf{CZF}) + (\mathbf{DNSP})$ *is sound.*

The type theory $\mathbf{ML}(\mathbf{CZF}) + (\mathbf{REM})$ and the set theory $\mathbf{CZF} + \mathbf{REM}$ are mutually interpretable. Recall that the set theory $\mathbf{CZF} + \mathbf{REM}$ has proof theoretic strength at least above that of Bounded Zermelo set theory, which is obtained from Zermelo set theory by limiting the separation axiom scheme to restricted formulas. This is because the power set axiom is derivable in $\mathbf{CZF} + \mathbf{REM}$ and Bounded Zermelo set theory has a double-negation interpretation into its intuitionistic counterpart. That set theory is in fact a subsystem of $\mathbf{CZF} + \mathbf{REM}$. The addition of the type theoretic principle $(\mathbf{DNSP})$ to $\mathbf{ML}(\mathbf{CZF})$ pushes therefore the proof theoretic strength of the type theory above that of second-order arithmetic.

References

1. Peter Aczel, The Type Theoretic Interpretation of Constructive Set Theory, in: A. MacIntyre, L. Pacholski and J. Paris (eds.), *Logic Colloquium '77*, (North-Holland, Amsterdam, 1978).
2. Peter Aczel, The Type Theoretic Interpretation of Constructive Set Theory: Inductive Definitions, in: R.B. Barcan Marcus, G. J. W. Dorn and P. Weingartner (eds.) *Logic, Methodology and Philosophy of Science, VII*, (North-Holland, Amsterdam, 1986).
3. Peter Aczel, Notes on Constructive Set Theory, Draft manuscript. Available at `http://www.cs.man.ac.uk/~petera`, 1997.
4. Peter Aczel, On Relating Type Theories and Set Theories, in T. Altenkirch, W. Naraschewski and B. Reus (eds.) *Types for Proofs and Programs, Proceedings of Types '98*, SLNCS 1657, (1999).
5. Peter Aczel, The Russell-Prawitz Modality, *Mathematical Structures in Computer Science*, vol. 11, (2001) 1 – 14.
6. Peter Aczel and Michael Rathjen, Notes on Constructive Set Theory, Preprint no. 40, Institut Mittag-Leffler, The Royal Swedish Academy of Sciences, 2001. Available at `http://www.ml.kva.se`.
7. Thierry Coquand and Erik Palmgren, Intuitionistic Choice and Classical Logic, *Archive for Mathematical Logic*, vol. 39 (2000) 53 – 74.
8. Thierry Coquand, Giovanni Sambin, Jan Smith and Silvio Valentini, Inductively Generated Formal Topologies, Submitted for publication, 2000.

9. Harvey M. Friedman, The Consistency of Classical Set Theory Relative to a Set Theory with Intuitionistic Logic, *Journal of Symbolic Logic*, vol. 38 (1973) 315 – 319.
10. Nicola Gambino and Peter Aczel, Frame-Valued Semantics for Constructive Set Theory, Preprint no. 39, Institut Mittag-Leffler, The Swedish Royal Academy of Sciences, 2001. Available at `http://www.ml.kva.se`.
11. Robin Grayson, Forcing in Intuitionistic Theories without Power Set, *Journal of Symbolic Logic*, vol. 48 (1983) 670 – 682.
12. Robin Grayson, Heyting-valued models for Intuitionistic Set Theory, in M. Fourman, C. Mulvey and D.S. Scott (eds.) *Applications of Sheaves*, vol. 743, SLNM (1979).
13. Edward Griffor and Michael Rathjen, The Strength of Some Martin-Löf's Type Theories, *Archiv for Mathematical Logic* vol. 33, (1994) 347 – 385.
14. Peter T. Johnstone, *Stone Spaces*, (Cambridge University Press, Cambridge, 1982).
15. Per Martin-Löf, *Intuitionistic Type Theories*, (Bibliopolis, Napoli, 1984).
16. Saunders MacLane and Ieke Moerdijk, *Sheaves in Geometry and Logic. A First Introduction to Topos Theory*, (Springer, Berlin, 1992).
17. Ieke Moerdijk and Erik Palmgren, Wellfounded Trees in Categories, *Annals of Pure and Applied Logic*, vol. 104, (2000) 189 – 218.
18. Ieke Moerdijk and Erik Palmgren, Type Theories, Toposes and Constructive Set Theory: Predicative Aspects of AST, *Annals of Pure and Applied Logic*, to appear.
19. Michael Rathjen, Edward Griffor and Erik Palmgren, Inaccessibility in Constructive Set Theory and Type Theory, *Annals of Pure and Applied Logic*, vol. 94, (1998) 181 – 200.
20. Giovanni Sambin, Intuitionistic Formal Spaces, in D. Skordev (ed.) *Mathematical Logic and its Applications* (Plenum, New York, 1987), 187 – 204.
21. Anne Troelstra and Dirk van Dalen, *Constructivism in Mathematics*, Vol. 1, Studies in Logic No. 121, (North Holland, 1988).