THE GENERALISED TYPE-THEORETIC INTEPRETATION OF CONSTRUCTIVE SET THEORY

NICOLA GAMBINO AND PETER ACZEL

Abstract. We present a generalisation of the type-theoretic interpretation of constructive set theory into Martin-Löf type theory. The original interpretation treated logic in Martin-Löf type theory via the propositions-as-types interpretation. The generalisation involves replacing Martin-Löf type theory with a new type theory in which logic is treated as primitive. The primitive treatment of logic in type theories allows us to study reinterpretations of logic, such as the double-negation translation.

Contents

Introduction

On interpreting set theories in type theories. The type-theoretic interpretation of Constructive Zermelo-Frankel set theory, or CZF for short, provides an explicit link between constructive set theory and Martin-Löf type theory [1, 2, 3]. This interpretation is also a useful tool in the proof-theoretical investigations of constructive formal systems [15] and allows us to relate the set-theoretic and type-theoretic approaches to the development of constructive mathematics [5, 25].

Date: June 16, 2005.
2000 *Mathematics Subject Classification.* 03F25 (Relative consistency and interpretations), 03F50 (Metamathematics of constructive systems).
Key words and phrases. Constructive set theory, Dependent type theory.

The first main component of the original type-theoretic interpretation of CZF is the propositions-as-types interpretation of logic. Under this interpretation, arbitrary formulas of CZF are interpreted as types, and restricted formulas as small types. By a small type we mean here a type represented by an element of the type universe that is part of the type theory in which CZF is interpreted. The propositions-as-types representation of logic plays a role in proving the validity of three schemes of CZF, namely Restricted Separation, Strong Collection, and Subset Collection. Validity of Restricted Separation follows from the representation of restricted propositions as small types, while the validity of both Strong Collection and Subset Collection follow from the type-theoretic axiom of choice, that holds in the propositions-as-types interpretation of logic [21]. The second main component of the type-theoretic interpretation is the definition of a type V, called the type of iterative sets, that is used to interpret the universe of sets of CZF. In this way, it is possible to obtain a valid interpretation of CZF in the Martin-Löf type theory $\mathrm{ML}_1 + \mathrm{W}$, which has rules for the usual forms of type, for a type universe reflecting these forms of type, and for W-types (see Section 1 for details).

Our aim here is to present a new type-theoretic interpretation of CZF. The novelty lies in replacing the pure type theory like $\mathrm{ML}_1 + \mathrm{W}$ with a suitable logic-enriched type theory. By a logic-enriched intuitionistic type theory we mean an intuitionistic type theory like $\mathrm{ML}_1 + \mathrm{W}$ that is extended with judgement forms to express, relative to a context of variable declarations, the notion of proposition and assertions that one proposition follows from others. Logic-enriched type theories have a straighforward interpretation in their pure counterparts that is obtained by following the propositions-as-types idea. When formulating logic-enriched type theories that extend pure type theories with rules for a type universe, like $\mathrm{ML}_1 + \mathrm{W}$, it is natural to have rules for a proposition universe to match the type universe. Elements of the proposition universe should be thought of as representatives for propositions whose quantifiers range over small types.

The new interpretation generalises the original type-theoretic interpretation in that logic is treated as primitive and not via the propositions-as-types interpretation. In particular, we will introduce a logic-enriched type theory, called ML(COLL), that has two collection rules, corresponding to the collection axiom schemes of CZF. Within the type theory ML(COLL) we define a type $\mathbb{V}$, called the type of iterative small classes, that can be used to interpret the universe of sets of CZF. The particular definition of $\mathbb{V}$ allows us to prove the validity of Restricted Separation without assuming the propositions-as-types interpretation of logic, and the collection rules of ML(COLL) allow us to prove the validity of Strong Collection and Subset Collection. We will therefore obtain a type-theoretic interpretation of CZF that does not rely on the propositions-as-types treatment of logic and in particular avoids any use of the type-theoretic axiom of choice.

The original interpretation and reinterpretations of logic. A first reason for the interest in the generalised interpretation is that it allows us to provide an analysis of the original interpretation. Indeed, we will show how the original interpretation can be viewed as the result of composing the generalised interpretation with a propositions-as-types interpretation. This analysis is obtained by considering a logic-enriched type theory $\mathrm{ML}(\mathrm{AC}+\mathrm{PU})$ with special rules expressing the axiom of choice and a correspondence between the proposition and type universes. These rules are valid under the proposition-as-types interpretation, so that $\mathrm{ML}(\mathrm{AC}+\mathrm{PU})$ can be intepreted in the pure type theory $\mathrm{ML}_1+\mathrm{W}$. Furthermore, the collection rules of $\mathrm{ML}(\mathrm{COLL})$ follow from the special rules of $\mathrm{ML}(\mathrm{AC}+\mathrm{PU})$, and therefore it is possible to view the generalised interpretation of CZF as taking place in $\mathrm{ML}(\mathrm{AC}+\mathrm{PU})$. We will then prove that the original interpretation of CZF in $\mathrm{ML}_1+\mathrm{W}$ can be seen as the composition of the generalised interpretation of CZF in $\mathrm{ML}(\mathrm{AC}+\mathrm{PU})$ followed by the propositions-as-types of $\mathrm{ML}(\mathrm{AC}+\mathrm{PU})$ into $\mathrm{ML}_1+\mathrm{W}$.

Another goal of this paper is to describe how logic-enriched type theories with collection rules, like $\mathrm{ML}(\mathrm{COLL})$, have a key advantage over logic-enriched type theories with the axiom of choice, like $\mathrm{ML}(\mathrm{AC}+\mathrm{PU})$. The advantage is that the former can accomodate reinterpretations of their logic, while the latter cannot. We focus our attention on reinterpretations of logic as determined by a map j that satisfies a type-theoretic version of the properties of a Lawvere-Tierney local operator in an elementary topos [19] or of a nucleus on a frame [17]. We will call such a j a local operator, and the reinterpretation of logic determined by it will be called the j-interpretation. A typical example of such an operator is provided by double-negation.

In our development, we consider initially a subsystem $\mathrm{ML}(\mathrm{COLL}^-)$ of $\mathrm{ML}(\mathrm{COLL})$, and $\mathrm{ML}(\mathrm{COLL})$ 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 local operator j, while the Subset Collection rule is not. In order to obtain the derivability of the j-interpretation of the Subset Collection rule, we will introduce a natural further assumption. We end the paper by describing a type-theoretic version of the double-negation translation, and give a proof-theoretic application.

Related work. The generalised type-theoretic interepretation is related to the investigations on categorical models for constructive set theories [22, 23, 13]. Indeed, one of our initial motivations was to study whether it was possible to obtain a type-theoretic version of the results in [23] concerning the interpretation of CZF in categories whose internal logic does not satisfy the axiom of choice. One essential difference, however, between the development presented here and the categorical results existing in most of the literature is that propositions and types interact in logic-enriched type theories in a more restricted way than in categories when logic is treated assuming the proposition-as-subobjects approach. In particular, logic-enriched type theories do not generally have rules that allow us to form types by separation, something that is instead a direct consequence of the proposition-as-subobjects representation of logic in categories [22, 23]. The category theoretic counterpart to the logic enrichment of a pure dependent type theory is roughly a first order fibration over a category that is already the base category of a fibration representing the dependent type theory. See, for example, [16, Chapter 11].

Extensions of pure type theories that allow the formation of types by separation have already been studied. One approach is via minimal type theories [20]. Minimal type theories may be understood as extensions of logic-enriched type theories with extra rules asserting that each proposition represents a type, that is to be thought of as the types of proofs of the proposition. Using these rules and the standard rules for Σ-types of the underlying pure type theory, the formation of types by separation can be easily obtained. Another approach is offered by the pure type theories with bracket types [6]. These are obtained by extending pure type theories with rules for a new form of type, called bracket type, that allows the representation of propositions as types with at most one element. This approach provides essentially a type-theoretic version of the proposition-as-subobjects idea. In the development of the generalised type-theoretic interpretation we preferred however to work within logic-enriched type theories, and avoid the assumption of extra rules allowing formation of types by separation. In this respect, the generalised type-theoretic interpretation given here is more general than the existing categorical models for CZF.

Constructive Set Theory. Let us conclude these introductory remarks by reviewing briefly the axiom system of the constructive set theory CZF, originally formulated to support the development of constructive mathematics in a set-theoretic framework [5]. For proof-theoretical investigations on constructive set theories we invite the reader to refer to [12, 18, 26, 27, 28, 29]. The axioms of CZF will be presented here in an extension of the language of first-order logic with primitive restricted quantifiers $(\forall x \in \alpha)$ and $(\exists x \in \alpha)$. The membership relation can then be defined by letting

$$\alpha \in \beta =_{\mathrm{def}} (\exists x \in \beta)(x = \alpha)$$

A formula is said to be *restricted* if all the quantifiers in it are restricted. We use letters $u, v, z, x, y, w, \dots$ for variables of the language, and greek letters $\alpha, \beta, \gamma, \dots$ to denote sets. Other greek letters are used to denote formulas. For formulas ϕ and ψ, we write $\phi \supset \psi$ for their implication and define $\phi \equiv \psi =_{\mathrm{def}} (\phi \supset \psi) \wedge (\psi \supset \phi)$.

The axiom system of CZF includes both logical and set-theoretic axioms and schemes. We will refer to axioms and schemes collectively as axiom schemes The logical axioms schemes include the standard ones for intuitionistic logic with equality and the following axiom schemes for restricted quantifiers:

$$(\forall x \in \alpha)\phi \equiv (\forall x)(x \in \alpha \supset \phi)\,, \qquad (\exists x \in \alpha)\phi \equiv (\exists x)(x \in \alpha \wedge \phi)$$

The set-theoretic axiom schemes of CZF can be conceptually divided into three groups: structural, basic set existence, and collection. The structural axiom schemes of CZF are Extensionality (1) and Set Induction (2).

$$(\forall x)(x \in \alpha \equiv x \in \beta) \supset (\alpha = \beta) \tag{1}$$

$$(\forall x)\big((\forall y \in x)\phi[y/x] \supset \phi\big) \;\supset\; (\forall x)\phi \tag{2}$$

The Extensionality axiom asserts that if two sets have the same elements, then they are equal. The Set Induction scheme is the intuitionistic counterpart of the classical Foundation axiom, and it is formulated as a scheme in which ϕ is an arbitrary formula. The first basic set existence axioms of CZF are Pairing (3), Union (4), and Infinity (5).

$$(\exists u)(\forall x)(x \in u \;\equiv\; (x = \alpha \vee x = \beta)) \tag{3}$$

$$(\exists u)(\forall x)(x \in u \equiv (\exists y \in \alpha)(x \in y)) \tag{4}$$

$$(\exists u)((\exists x)(x \in u) \wedge (\forall x \in u)(\exists y \in u)(x \in y)) \tag{5}$$

They are formulated as in classical set theory. A further basic set existence scheme of CZF is Restricted Separation. It is the scheme in (6), where θ is a restricted formula.

$$(\exists u)(\forall x)(x \in u \equiv x \in \alpha \wedge \theta) \tag{6}$$

The Restricted Separation scheme is a weakening of the classical scheme of Full Separation, obtained by limiting the kind of formulas allowed in the scheme. To formulate the two collection schemes of CZF we use the abbreviation

$$(\forall\exists\ \tfrac{x\in\alpha}{y\in\beta})\,\phi =_{\mathrm{def}} (\forall x \in \alpha)(\exists y \in \beta)\phi \wedge (\forall y \in \beta)(\exists x \in \alpha)\phi$$

where ϕ is an arbitrary formula. Note that in the formula

$$(\forall\exists\ \tfrac{x\in\alpha}{y\in\beta})\,\phi$$

free occurrences of x and y in ϕ get bound by the operator $(\forall\exists\ \tfrac{x\in\alpha}{y\in\beta})$. The Strong Collection (7) and Subset Collection (8) schemes are given below.

$$(\forall x \in \alpha)(\exists y)\phi \supset (\exists u)(\forall\exists\ \tfrac{x\in\alpha}{y\in u})\,\phi \tag{7}$$

$$(\exists v)(\forall z)[(\forall x \in \alpha)(\exists y \in \beta)\phi \supset (\exists u \in v)(\forall\exists\ \tfrac{x\in\alpha}{y\in u})\,\phi] \tag{8}$$

Note that in the Subset Collection scheme (8) the formula ϕ may have free occurrences of z which get bound by the universal quantifier $(\forall z)$. The Strong Collection scheme is a mild strengthening of the Collection scheme, which in turn implies the Replacement scheme. The strengthening is appropriate to the setting of a set theory without the Full Separation scheme [9].

The Subset Collection scheme is instead a weakening of the Power Set axiom and a strengthening of the Exponentiation axiom, which asserts that the class of functions between two sets is again a set [24]. In [18] it is shown that, in the presence of axioms (1) – (7), the Subset Collection scheme is independent of the Exponentiation axiom.

Outline of the paper. A review of Martin-Löf pure type theories is presented in Section 1, which also serves to fix the notation used in the reminder of the paper. The list of rules for the type theories used here is given in Appendix A. Logic-enriched type theories are introduced in Section 2, where we also describe their propositions-as-types interpretation. In Section 3 we define the generalised type-theoretic interpretation of CZF. The relationship between the original and the generalised type-theoretic interpretations is then described in Section 4. Section 5, finally, describes the reinterpretations of logic.

Note. Logic-enriched type theories were originally introduced in [4], where the results given here were announced without proofs. Here we give complete proofs of the results concerning the generalised type-theoretic interpretation of constructive set theory. We hope to present more material on logic-enriched type theories in the future. In particular, we are planning a sequel to this paper which will contain further results concerning the propositions-as-types interpretation of logic-enriched type theories into their pure counterparts.

Acknowledgements. The first author wishes to thank the Department of Computer Science, University of Manchester, where most of the research described here was conducted.

1. Pure type theories

Standard pure type theories. A standard pure type theory has judgements of form $(\Gamma)\ \mathcal{B}$, 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 following forms.

$$A : \mathrm{type} \qquad A = A' : \mathrm{type} \qquad a : A \qquad a = a' : A \tag{9}$$

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

$$(\ )\ A_1 : \mathrm{type} \quad (x_1 : A_1)\ A_2 : \mathrm{type} \quad \ldots \quad (x_1 : A_1, \ldots, x_{n-1} : A_{n-1})\ A_n : \mathrm{type}$$

are derivable. The well-formedness of the forms of judgement in (9) has other presuppositions: in a well-formed context Γ, the judgement $A = A' : \mathrm{type}$ presupposes that $A : \mathrm{type}$ and $A' : \mathrm{type}$, the judgement $a : A$ presupposes that $A : \mathrm{type}$, and the judgement $a = a' : A$ presupposes that $a : A$ and $a' : A$. In the rest of the paper we prefer to leave out the empty context whenever possible, so that $(\ )\ A : \mathrm{type}$ will be written simply as $A : \mathrm{type}$.

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 convenient to suppress mention of a context that is common to both the premisses and the conclusion of the rule. For example we write the reflexivity rule for type equality as just

$$\frac{A : \mathrm{type}}{A = A : \mathrm{type}}$$

but when we apply this rule we are allowed to infer $(\Gamma)\ A = A : \mathrm{type}$ from $(\Gamma)\ A : \mathrm{type}$ for any well-formed context Γ.

Martin-Löf type theory. We use ML to stand for a variant of Martin-Löf's standard type theory without universes or W-types [21, 25]. 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. We do not assume binary sums as primitive but define them. To do so, we allow dependent types to be defined by cases on $\mathbb{2}$ as follows: under the assumption that $A_1 : \mathrm{type}$ and $A_2 : \mathrm{type}$ we allow the formation of $\mathbb{R}_2(c, A_1, A_2) : \mathrm{type}$ whenever $c : \mathbb{2}$. Furthermore there are rules stating that the judgements

$$\mathbb{R}_2(1_2, A_1, A_2) = A_1 : \mathrm{type} \qquad \mathbb{R}_2(2_2, A_1, A_2) = A_2 : \mathrm{type}$$

are derivable. Here $1_2 : \mathbb{2}$ and $2_2 : \mathbb{2}$ are the canonical elements of the type $\mathbb{2}$. This form of type allows us to define binary sums. For types A_1 and A_2 we define

$$A_1 + A_2 =_{\mathrm{def}} (\Sigma z : \mathbb{2})\mathbb{R}_2(z, A_1, A_2)$$

Binary product and function types are defined as usual

$$A_1 \times A_2 =_{\mathrm{def}} (\Sigma\_ : A_1)A_2 \qquad A_1 \to A_2 =_{\mathrm{def}} (\Pi\_ : A_1)A_2$$

Here and in the following the symbol _ indicates an anonymous bound variable. To fix notation let us also recall that there are derivable rules expressing the first and second projection of an element of a Σ-type, as follows

$$\frac{c : (\Sigma x : A)B}{c.1 : A} \qquad \frac{c : (\Sigma x : A)B}{c.2 : B[c.1/x]} \tag{10}$$

In summary, the primitive forms of type of ML are

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

Table 1 presents the raw syntax for the expressions of ML that we are going to use throughout the paper. For the convenience of the readers, the complete set of rules of ML is recalled in Appendix A.

Form of type	Canonical expression	Eliminating expression
$\mathbb{0}$		$\mathrm{r}_0(e)$
$\mathbb{1}$	0_1	$\mathrm{r}_1(e, c)$
$\mathbb{2}$	$1_2,\ 2_2$	$\mathrm{r}_2(e, c_1, c_2)$
$\mathbb{N}$	$0,\ \mathrm{succ}(e)$	$\mathrm{r}_{\mathbb{N}}(e, c, (x, y)d)$
$(\Sigma x : A)B$	$\mathrm{pair}(a, b)$	$\mathrm{split}(e, (x, y)c)$
$(\Pi x : A)B$	$(\lambda x : A)b$	$\mathrm{app}(f, a)$

TABLE 1. Types and expressions of the ML type theory.

Wellfounded trees. Types of wellfounded trees, or W-types for short, play a crucial role in the intrerpretation of constructive set theories in dependent type theories. Let us then briefly review the rules concerning this form of type. The formation rule and introduction rules for W-types are the following

$$\frac{A : \mathrm{type} \qquad (x : A)\ B : \mathrm{type}}{(\mathbb{W}x : A)B : \mathrm{type}} \qquad \frac{a : A \qquad t : B[a/x] \to (\mathbb{W}x : A)B}{\mathrm{sup}(a, t) : (\mathbb{W}x : A)B}$$

A canonical element $\mathrm{sup}(a, t)$ of $W =_{\mathrm{def}} (\mathbb{W}x : A)B$ should be thought of as the tree with a root labelled by $\mathrm{sup}(a, t)$. The branches departing from the root are indexed by elements of $B[a/x]$ with nodes labelled by the elements $\mathrm{app}(t, b)$ for $b : B[a/x]$.

In the formulation of the elimination and computation rules, we suppress mention of the judgement $(z : W)\ C : \mathrm{type}$ that is part of the premisses. Let $(\Gamma) =_{\mathrm{def}} (x : A\,,\ u : B \to W\,,\ v : (\Pi y : B)C[\mathrm{app}(u, y)/z])$ so that we can write the elimination rule as

$$\frac{e : W \qquad (\Gamma)\ c : C[\mathrm{sup}(x, u)/z]}{\mathrm{r_W}(e, (x, u, v)c) : C[e/z]}$$

and the computation rule as

$$\frac{a : A \qquad t : B[a/x] \to W \qquad (\Gamma)\ c : C[\mathrm{sup}(x, u)/z]}{\mathrm{rec}(\mathrm{sup}(a, t)) = c[a, t, (\lambda y : B[a/x])\mathrm{rec}(\mathrm{app}(t, y))/x, u, v] : C[\mathrm{sup}(a, t)/z]}$$

where $\mathrm{rec}(e) =_{\mathrm{def}} \mathrm{r_W}(e, (x, u, v)c)$, for $e : W$.

Type universes. We will consider type theories that include rules for a type universe $\mathbb{U}$ of small types, or rather of representatives for small types. We adopt a slight variant of the so-called Tarski-style formulation of type universes, that we now present. The type universe has formation rule

$$\mathbb{U} : \mathrm{type}$$

For each $a : \mathbb{U}$, we write $\mathbb{T}(a)$: type for the small type represented by a. Therefore, the elimination rule for the type universe is stated as

$$\frac{a : \mathbb{U}}{\mathbb{T}(a) : \text{type}}$$

The introduction and computation rules for the type universe express that $\mathbb{U}$ reflects all the forms of type of ML. For example, to express that $\mathbb{U}$ reflects the type $\mathbb{O}$ we have the introduction rule

$$\widehat{\mathbb{O}} : \mathbb{U}$$

which has an associated computation rule asserting

$$\mathbb{T}(\widehat{\mathbb{O}}) = \mathbb{O} : \text{type}$$

To reflect Σ-types, there is the introduction rule

$$\frac{a : \mathbb{U} \quad (x : \mathbb{T}(a))\ b : \mathbb{U}}{(\Sigma x : a) b : \mathbb{U}}$$

and its associated computation rule

$$\mathbb{T}((\Sigma x : a)b) = (\Sigma x : \mathbb{T}(a))\, \mathbb{T}(b) : \text{type}$$

The complete set of rules for the type universe is given in Appendix A. Note that the symbol Σ appears both in the judgement $(\Sigma x : a)b : \mathbb{U}$ and in the judgement $(\Sigma x : \mathbb{T}(a))\, \mathbb{T}(b)$: type. Similar notation is used to reflect the type formation rules of ML in the type universe. Since the forms of judgement in which the same symbol appears are different, there is no reason for confusion.

We write MLW and ML_1 for the type theories that are obtained from ML by adding rules for W-types and rules for a type universe, respectively. The type theory MLW_1 is like ML_1 except that the rules for the W-types are added and the type universe $\mathbb{U}$ also reflects W-types. A particularly important role will be played in the following by the natural subtheory $\mathrm{ML}_1 + \mathrm{W}$ of MLW_1, which has W-types but they are not reflected in $\mathbb{U}$. Many proof-theoretical results concerning these pure type theories are presented in [15].

2. Logic-enriched type theories

Adding Predicate Logic. Given a standard pure type theory we may consider extending it with 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 forms

$$\phi : \text{prop} \qquad\qquad \phi_1, \ldots, \phi_m \Rightarrow \phi$$

These judgements express, relative to the context (Γ), that ϕ is a proposition and that ϕ follows from $\phi_1, \ldots, \phi_m$, respectively. In the context Γ, the well-formedness of the judgement $\phi_1, \ldots, \phi_m \Rightarrow \phi$ presupposes that ϕ_i : prop (for $i = 1, \ldots, m$) and ϕ : 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. For example, the rules for the existential quantifier are as follows.

$$\frac{A : \text{type} \qquad (x : A)\ \phi : \text{prop}}{(\exists x : A)\ \phi : \text{prop}} \qquad \frac{(x : A)\ \phi : \text{prop} \qquad a : A \qquad \phi[a/x]}{(\exists x : A)\phi}$$

$$\frac{(\exists x : A)\phi \qquad \psi : \text{prop} \qquad (x : A)\ \phi \Rightarrow \psi}{\psi}$$

The negation $\neg\phi$ of a proposition ϕ is defined by letting $\neg\phi =_{\text{def}} \phi \supset \bot$, and logical equivalence is expressed with the proposition

$$\phi \equiv \psi =_{\text{def}} (\phi \supset \psi) \wedge (\psi \supset \phi) \tag{11}$$

where ϕ and ψ are propositions. 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 $\Rightarrow \phi$.

Induction Rules. It is possible 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. For each inductive type C of MLW, we have an induction rule of the form

$$\frac{(z : C)\ \phi : \text{prop} \qquad e : C \qquad \mathit{IND}_C}{\phi[e/z]}$$

where the correspondence between the form of C and the premisses IND_C is described in Table 2.

C	IND_C
$\mathbb{0}$	
$\mathbb{1}$	$\phi[0_1/z]$
$\mathbb{2}$	$\phi[1_2/z]\,, \phi[2_2/z]$
$\mathbb{N}$	$\phi[0/z]\,, (x : \mathbb{N})\ \phi[x/z] \Rightarrow \phi[\text{succ}(x)/z]$
$(\Sigma x : A)B$	$(x : A, y : B)\ \phi[\text{pair}(x, y)/z]$
$(Wx : A)B$	$(x : A, u : B \to C)\ (\forall y : B)\phi[\text{app}(u, y)/z] \Rightarrow \phi[\text{sup}(x, u)/z]$

TABLE 2. Inductive types and premisses of their induction rules.

The proposition universe. When adding logic to a standard pure type theory T that includes 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

$$\mathbb{P} : \mathrm{type}$$

The rules for $\mathbb{P}$ express that elements of this type are to be thought of as representatives for propositions whose quantifiers range over small types. Indeed, the elimination rule

$$\frac{p : \mathbb{P}}{\tau(p) : \mathrm{prop}}$$

expresses that each object $p : \mathbb{P}$ represents a proposition. To express that the false proposition has a representative in $\mathbb{P}$ we have the introduction rule

$$\widehat{\bot} : \mathbb{P}$$

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 defined in (11). For example, the rule relative to the false proposition is

$$\tau(\widehat{\bot}) \equiv \bot$$

Similarly, existential quantification over small types is reflected in $\mathbb{P}$ with the introduction rule

$$\frac{a : \mathbb{U} \qquad (x : \mathbb{T}(a))\ p : \mathbb{P}}{(\exists x : a)p : \mathbb{P}}$$

and the rule

$$\tau((\exists x : a)p) \equiv (\exists x : \mathbb{T}(a))\tau(p)$$

The set of rules for the proposition universe $\mathbb{P}$ is presented in Appendix A.

Some logic-enriched type theories. Given a pure type theory T, we write $\mathrm{T} + \mathrm{IL}$ for the logic-enriched type theory that is obtained from T by adding rules for predicate logic. We then write $\mathrm{T} + \mathrm{IL} + \mathrm{IND}$ for the logic-enriched type theory that has induction rules for each of the forms of inductive type of T. When the pure type theory T includes ML_1 then we write $\mathrm{T} + \mathrm{IL}_1$ for the enrichment of T with intuitionistic predicate logic and also the rules for $\mathbb{P}$, and $\mathrm{T} + \mathrm{IL}_1 + \mathrm{IND}$ for its extension with induction rules.

In the following we will be interested in an extension of the logic-enriched type theory $\mathrm{ML}_1 + \mathrm{W} + \mathrm{IL}_1 + \mathrm{IND}$, whose primitive forms of type are the following.

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

There are predicate logic rules, induction rules for $\mathbb{0}$, $\mathbb{1}$, $\mathbb{2}$, $\mathbb{N}$, $(\Sigma x : A)B$ and $(\mathbb{W}x : A)B$, and rules for the type $\mathbb{P}$. We will work informally in the logic-enriched type theory $\mathrm{ML}_1 + \mathrm{W} + \mathrm{IL}_1 + \mathrm{IND}$ in Section 3.

Propositions-as-types. The logic-enriched type theory $\mathrm{ML} + \mathrm{IL}$ has a straightforward interpretation into the pure type theory ML that is obtained by following the propositions-as-types idea. Each induction rule can also be justified under the propositions-as-types intepretation of logic by using the elimination rule of the inductive type to which the induction rule is associated. Furthermore, the derivability of the type-theoretic axiom of choice [21] implies that the rule

$$\frac{A : \mathrm{type} \qquad (x : A)\ B : \mathrm{type} \qquad (x : A, y : B)\ \phi : \mathrm{prop}}{(\forall x : A)(\exists y : B)\phi \supset \big(\exists u : (\Pi x : A)B\big)(\forall x : A)\phi[\mathrm{app}(u, x)/y]} \tag{AC}$$

has a valid propositions-as-types interpretation in the pure type theory ML. Hence, the propositions-as-types interpretation reduces the logic-enriched type theory $\mathrm{ML} + \mathrm{IL} + \mathrm{IND} + \mathrm{AC}$ to the pure type theory ML. An analogous result holds when we add rules for W-types on both sides.

The propositions-as-types idea extends to logic-enriched type theories with a proposition universe. The extension is obtained by interpreting the proposition universe $\mathbb{P}$ as the type universe $\mathbb{U}$, and representatives for small propositions as representatives for small types, again following the propositions-as-types idea. The logic-enriched type theory $\mathrm{ML}_1 + \mathrm{IL}_1$ has then an interpretation into the pure type theory ML_1. Since the rules for the proposition universe $\mathbb{P}$ are completely analogous to the rules for the type universe $\mathbb{U}$ it is possible to see that also the proposition

$$(\forall p : \mathbb{P})(\exists u : \mathbb{U})\big(\tau(p) \equiv (\exists_{\_} : \mathbb{T}(u))\top\big) \tag{PU}$$

is interpreted as an inhabited type by this extension of the propositions-as-types interpretation. As a consequence of these facts, the logic-enriched type theory $\mathrm{ML}(\mathrm{AC} + \mathrm{PU}) =_{\mathrm{def}} \mathrm{ML}_1 + \mathrm{W} + \mathrm{IL}_1 + \mathrm{IND} + \mathrm{AC} + \mathrm{PU}$ admits an interpretation into the pure type theory $\mathrm{ML}_1 + \mathrm{W}$. In Section 3 we will introduce collection principles and show that the rules (AC) and (PU) allow us to derive them. The following lemma will be helpful to do so.

Lemma 2.1. *Assuming* (AC) *and* (PU) *there exists* $\mathrm{t} : \mathbb{P} \to \mathbb{U}$ *such that, for* $p : \mathbb{P}$, *the judgement* $\tau(p) \equiv (\exists_{\_} : \mathbb{T}(\mathrm{app}(\mathrm{t}, p))\top$ *is derivable.*

Proof. To prove the claim, it is suffient to apply (AC) to (PU). ☐

3. The generalised type-theoretic interpretation

Some notions of collection. Given a type A what is a *collection* of objects of type A? We may consider three approaches to this question: logical, combinatorial, and hybrid. The logical approach is to take a collection to be a class on A, i.e. a propositional function $(x : A)\ \phi : \mathrm{prop}$. The objects of such a collection are the $a : A$ such that $\phi[a/x]$ holds. By contrast the combinatorial approach is to take a collection to be a family $(x : I)\ b : A$ indexed by a type I. This time the objects are the $b[i/x] : A$ for $i : I$. Finally, the hybrid approach takes a collection to consist of a pair given by a family $(x : I)\ b : A$ and a class $(x : I)\ \phi : \mathrm{prop}$ on the index type I. The objects of such a collection are the $b[i/x] : A$ for those $i : I$ such that $\phi[i/x]$ holds.

A problem with each of these notions of collection is that we cannot generally expect there to be a type of all collections of objects of type A for all types A. What we can expect is to be able to form types of *small collections* using the type and proposition universes. For example, the type of small collections for the logical approach is given by

$$\mathrm{Cla}(A) =_{\mathrm{def}} A \to \mathbb{P}$$

We get the following types of small collections for the combinatorial and hybrid notions of small collection of objects of type A. We shall refer to these types as the types of *small families* and of *small subclasses* of A, respectively:

$$\mathrm{Fam}(A) =_{\mathrm{def}} (\Sigma x : \mathbb{U})(\mathbb{T}(x) \to A) \tag{12}$$

$$\mathrm{Sub}(A) =_{\mathrm{def}} (\Sigma x : \mathbb{U})\big((\mathbb{T}(x) \to \mathbb{P}) \times (\mathbb{T}(x) \to A)\big) \tag{13}$$

The properties of these types play an essential role in the development of the original and generalised type-theoretic interpretations of CZF, and it is therefore convenient to introduce some notation to manipulate efficiently their elements. We do so by exploiting the notation for projections of elements of Σ-types as given in (10). Given $\alpha : \mathrm{Fam}(A)$ define

$$\mathrm{el}(\alpha) =_{\mathrm{def}} \alpha.1 : \mathbb{U}$$

We will write $\mathrm{el}(\alpha)$: type to denote also the small type $\mathbb{T}(\mathrm{el}(\alpha))$, associated with $\mathrm{el}(\alpha) : \mathbb{U}$, since the judgement makes clear whether we are considering a small type or the element in the type universe that represents it. We deal analogously with the elements in type and proposition universes that we define below. For $x : \mathrm{el}(\alpha)$, let

$$\mathrm{val}(\alpha, x) =_{\mathrm{def}} \mathrm{app}(\alpha.2, x) : A$$

Using the definitions introduced above, we can form propositions by quantifying over a small family, i.e. if $(x : A)\ \phi$: prop and $\alpha : \mathrm{Fam}(A)$ we can define the propositions $(\forall x \in \alpha)\phi$ and $(\exists x \in \alpha)\phi$ as follows.

$$\begin{aligned} (\forall x \in \alpha)\ \phi &=_{\mathrm{def}} (\forall x : \mathrm{el}(\alpha))\ \phi[\mathrm{val}(\alpha, x)/x] : \mathrm{prop} \\ (\exists x \in \alpha)\ \phi &=_{\mathrm{def}} (\exists x : \mathrm{el}(\alpha))\ \phi[\mathrm{val}(\alpha, x)/x] : \mathrm{prop} \end{aligned} \tag{14}$$

In a similar way, it is possible to form elements of the proposition universe by quantifying over a small family. For $(x : A)\ p : \mathbb{P}$ we define

$$\begin{aligned} (\forall x \in \alpha)\ p &=_{\mathrm{def}} (\forall x : \mathrm{el}(\alpha))\ p[\mathrm{val}(\alpha, x)/x] : \mathbb{P} \\ (\exists x \in \alpha)\ p &=_{\mathrm{def}} (\exists x : \mathrm{el}(\alpha))\ p[\mathrm{val}(\alpha, x)/x] : \mathbb{P} \end{aligned} \tag{15}$$

The notation introduced in (14) and (15) does not lead to confusion, since the rules for the proposition universe $\mathbb{P}$ imply that the judgements

$$\begin{aligned} (\forall x \in \alpha)\, \tau(p) &\equiv \tau\big((\forall x \in \alpha)p\big) \\ (\exists x \in \alpha)\, \tau(p) &\equiv \tau\big((\exists x \in \alpha)p\big) \end{aligned} \tag{16}$$

are derivable. We now develop an analogous system of abbreviations for the type of small subclasses of a type, as defined in (13). For $\alpha : \mathrm{Sub}(A)$ we define

$$\mathrm{el}(\alpha) =_{\mathrm{def}} \alpha.1 : \mathbb{U}$$

For $x : \mathrm{el}(\alpha)$ we define

$$\begin{aligned} \mathrm{dom}(\alpha, x) &=_{\mathrm{def}} \mathrm{app}(\alpha.2.1, x) : \mathbb{P} \\ \mathrm{val}(\alpha, x) &=_{\mathrm{def}} \mathrm{app}(\alpha.2.2, x) : A \end{aligned}$$

For $(x : A)\ \phi : \mathrm{prop}$ we define

$$\begin{aligned} (\forall x \in \alpha)\ \phi &=_{\mathrm{def}} (\forall x : \mathrm{el}(\alpha))\ (\mathrm{dom}(\alpha, x) \supset \phi[\mathrm{val}(\alpha, x)/x]) : \mathrm{prop} \\ (\exists x \in \alpha)\ \phi &=_{\mathrm{def}} (\exists x : \mathrm{el}(\alpha))\ (\mathrm{dom}(\alpha, x) \wedge \phi[\mathrm{val}(\alpha, x)/x]) : \mathrm{prop} \end{aligned} \tag{17}$$

Finally, for $(x : A)\ p : \mathbb{P}$ we define

$$\begin{aligned} (\forall x \in \alpha)\ p &=_{\mathrm{def}} (\forall x : \mathrm{el}(\alpha))\ (\mathrm{dom}(\alpha, x) \supset p[\mathrm{val}(\alpha, x)/x]) : \mathbb{P} \\ (\exists x \in \alpha)\ p &=_{\mathrm{def}} (\exists x : \mathrm{el}(\alpha))\ (\mathrm{dom}(\alpha, x) \wedge p[\mathrm{val}(\alpha, x)/x]) : \mathbb{P} \end{aligned} \tag{18}$$

Judgements analogous to those in (16) are easily derivable, so that the definitions in (17) and (18) are compatible.

Separation in type theory. The set theory CZF, like the systems of classical axiomatic set theory, allows the formation of sets of sets of sets of ... but does not allow non-well-founded sets. So to interpret the universe of sets of CZF as a type in type theory we need a type of iterative sets obtained by inductively iterating some notion of 'set of'. When we use the combinatorial approach to interpret the notion of 'set of' there is a problem with the justification of the Restricted Separation scheme of CZF if we wish to avoid the propositions-as-types representation of logic. The problem can be explained as follows. In CZF, for a set α and a restricted formula θ, Restricted Separation, as stated in (6), asserts that there is a set γ such that $(\forall x)\big(x \in \gamma \equiv (x \in \alpha \wedge \theta)\big)$. It is then straightforward to see that for a formula ϕ it holds

$$\begin{aligned} (\forall x \in \gamma)\ \phi &\equiv (\forall x \in \alpha)\ (\theta \supset \phi) \\ (\exists x \in \gamma)\ \phi &\equiv (\exists x \in \alpha)\ (\theta \wedge \phi) \end{aligned} \tag{19}$$

In the logic-enriched type theory $\mathrm{ML}(\mathrm{AC} + \mathrm{PU})$, given $\alpha : \mathrm{Fam}(A)$ and $(x : A)\ p : \mathbb{P}$, then we can define $\gamma : \mathrm{Fam}(A)$ such that for $(x : A)\ \phi : \mathrm{prop}$ we can derive judgements as in (19), but where $\theta =_{\mathrm{def}} \tau(p)$, for $x : \mathrm{el}(\alpha)$. The reason is that in $\mathrm{ML}(\mathrm{AC} + \mathrm{PU})$ it is possible to prove the existence of a function $\mathrm{t} : \mathbb{P} \to \mathbb{U}$ as in Lemma 2.1, and so to define $\gamma : \mathrm{Fam}(A)$ such that

$$\mathrm{el}(\gamma) = (\Sigma x : \mathrm{el}(\alpha))\, \mathrm{app}(\mathrm{t}, p[\mathrm{val}(\alpha, x)/x]) : \mathbb{U} \tag{20}$$

and

$$\mathrm{val}(\gamma, z) = \mathrm{val}(\alpha, z.1) : A \tag{21}$$

for $z : \mathrm{el}(\gamma)$. But without the assumption of (AC) and (PU), the proof of Lemma 2.1 cannot be carried out. To overcome this difficulty we adopt the hybrid notion of small collection instead of the combinatorial one. Note that the purely logical notion of small collection cannot be used to get any kind of iterative notion of set essentially because $\mathrm{Cla}(A)$ is not positive in A. So it seems that the use of the hybrid notion is the natural way to incorporate the necessary logical ingredient in the notion of small collection. We are therefore led to study the properties of small subclasses of a type.

Small subclasses. To define explicitly small subclasses of a type A it will be convenient to adopt the following convention. For derivable judgements of the form $a : \mathbb{U}$, $(x : \mathbb{T}(a))\ p : \mathbb{P}$, and $(x : \mathbb{T}(a))\ t : A$, we will define an element of $\mathrm{Sub}(A)$ by saying that it is the small subclass γ such that

$$\mathrm{el}(\gamma) = \mathbb{T}(a) : \mathrm{type} \tag{22}$$

and that, for $x : \mathbb{T}(a)$, the judgements

$$\begin{array}{c} \mathrm{dom}(\gamma, x) \equiv \tau(p) \\ \mathrm{val}(\gamma, x) = t : A \end{array} \tag{23}$$

are derivable. Note that if $\gamma = \delta : \mathrm{Sub}(A)$, where

$$\delta =_{\mathrm{def}} \mathrm{pair}(a, \mathrm{pair}((\lambda x : \mathbb{T}(a))p, (\lambda x : \mathbb{T}(a))t)) : \mathrm{Sub}(A)$$

then the judgements in (22) and (23) can actually be derived. We begin our discussion of set-theoretic constructs with the empty set. Recalling that $\mathbb{T}(\widehat{\mathbb{O}}) = \mathbb{O} : \mathrm{type}$, we define

$$\emptyset_A : \mathrm{Sub}(A)$$

to be the small subclass γ of A such that $\mathrm{el}(\gamma) = \mathbb{O} : \mathrm{type}$ and, for $x : \mathbb{O}$, the judgements $\mathrm{dom}(\gamma, x) \equiv \bot$ and $\mathrm{val}(\gamma, x) = \mathrm{r}_0(x) : A$ are derivable.

Lemma 3.1. *There exists $\gamma : \mathrm{Sub}(A)$ such that the judgements*

$$\begin{array}{ll} (\forall x \in \gamma)\ \phi & \equiv \top \\ (\exists x \in \gamma)\ \phi & \equiv \bot \end{array}$$

are derivable.

Proof. Let $\gamma = \emptyset_A : \mathrm{Sub}(A)$. The required conclusion follows directly by the definition of quantification over small subclasses in (17). □

To introduce the pairing operation, let us recall that $\mathbb{T}(\widehat{\mathbb{2}}) = \mathbb{2} : \mathrm{type}$. For $a_1, a_2 : A$ we can then define

$$\{a_1, a_2\} : \mathrm{Sub}(A)$$

as the small subclass γ of A such that $\mathrm{el}(\gamma) = \mathbb{2} : \mathrm{type}$ and, for $x : \mathbb{2}$, the judgements $\mathrm{dom}(\gamma, x) \equiv \top$ and $\mathrm{val}(\gamma, x) = \mathrm{r}_2(x, a_1, a_2) : A$ are derivable. The properties of this operation are stated in the next lemma.

Lemma 3.2. *Let $a_1, a_2 : A$. There exists $\gamma : \mathrm{Sub}(A)$ such that*

$$\begin{array}{lll} (\forall x \in \gamma)\ \phi & \equiv & \phi[a_1/x] \wedge \phi[a_2/x] \\ (\exists x \in \gamma)\ \phi & \equiv & \phi[a_1/x] \vee \phi[a_2/x] \end{array}$$

are derivable.

Proof. Let $\gamma = \{a_1, a_2\} : \mathrm{Sub}(A)$. Unfolding the definition of quantification over γ, we derive the following judgement.

$$(\forall x \in \gamma)\ \phi \ \equiv\ (\forall x : \mathbb{2})\ \phi[\mathrm{r}_2(a_1, a_2, x)/x]$$

Now, note that the following holds.

$$(\forall x : \mathbb{2})\ \phi[\mathrm{r}_2(a_1, a_2, x)/x] \ \equiv\ \phi[a_1/x] \wedge \phi[a_2/x]$$

The left-to-right implication is obtained with the $\forall$-elimination, and the right-to-left implication follows by the $\mathbb{2}$-induction rule. We have therefore obtained that $(\forall x \in \gamma)\ \phi$ is equivalent to $\phi[a_1/x] \wedge \phi[a_2/x]$ as required. We can derive the equivalence between $(\exists x \in \gamma)\ \phi$ and $\phi[a_1/x] \vee \phi[a_2/x]$ with an analogous reasoning: first unfold the definitions of restricted quantifiers, then use the $\mathbb{2}$-induction rule and the $\vee$-elimination rule. □

As a special case of the pairing operation defined above, we obtain the definition of singletons. For $a : A$, we define

$$\{a\} =_{\mathrm{def}} \{a, a\} : \mathrm{Sub}(A)$$

Lemma 3.3. *Let $a : A$. There exists $\gamma : \mathrm{Sub}(A)$ such that the judgements*

$$\begin{array}{rcl} (\forall x \in \gamma)\ \phi & \equiv & \phi[a/x] \\ (\exists x \in \gamma)\ \phi & \equiv & \phi[a/x] \end{array}$$

are derivable.

Proof. Let $\gamma =_{\mathrm{def}} \{a\} : \mathrm{Sub}(A)$. The proofs of the claims follow directly from Lemma 3.2. □

Let $\mathrm{Sub}^2(A) =_{\mathrm{def}} \mathrm{Sub}(\mathrm{Sub}(A))$. For $\alpha : \mathrm{Sub}^2(A)$ we define

$$\bigcup \alpha : \mathrm{Sub}(A)$$

as the small subclass γ of A such that $\mathrm{el}(\gamma) = (\Sigma y : \mathrm{el}(\alpha))\, \mathrm{el}(\mathrm{val}(\alpha, y))$ and, for $z : \mathrm{el}(\gamma)$, the judgements

$$\begin{array}{c} \mathrm{dom}(\gamma, z) \ \equiv\ \mathrm{dom}(\alpha, z.1) \wedge \mathrm{dom}(\mathrm{val}(\alpha, z.1), z.2) \\ \mathrm{val}(\gamma, z) = \mathrm{val}(\mathrm{val}(\alpha, z.1), z.2) : A \end{array}$$

are derivable. The next lemma shows that this operation has the properties of the set-theoretic union.

Lemma 3.4. *Let $\alpha : \mathrm{Sub}^2(A)$. There exists $\gamma : \mathrm{Sub}(A)$ such that the judgements*

$$\begin{array}{rcl} (\forall x \in \gamma)\ \phi & \equiv & (\forall y \in \alpha)(\forall x \in y)\ \phi \\ (\exists x \in \gamma)\ \phi & \equiv & (\exists y \in \alpha)(\exists x \in y)\ \phi \end{array}$$

are derivable.

Proof. Let $\gamma = \bigcup \alpha : \mathrm{Sub}(A)$. By the computation rules for Σ-types we obtain that for $y : \mathrm{el}(\alpha)$ and $x : \mathrm{el}(\mathrm{val}(\alpha, y))$

$$\mathrm{val}(\gamma, \mathrm{pair}(x, y)) = \mathrm{val}(\mathrm{val}(a, y), x) : A$$

holds. We want to show that $(\forall x \in \gamma)\phi$ is equivalent to $(\forall y \in \alpha)(\forall x \in y)\phi$. It is convenient to consider $\psi =_{\mathrm{def}} (\forall z : \mathrm{el}(\gamma))\, \mathrm{dom}(\gamma, z) \supset \phi[\mathrm{val}(\gamma, z)/x]$. Let $\theta =_{\mathrm{def}} \mathrm{dom}(\alpha, y) : \mathrm{prop}$, $\eta =_{\mathrm{def}} \mathrm{dom}(\mathrm{val}(\alpha, y), x) : \mathrm{prop}$, for $y : \mathrm{el}(\alpha)$ and $x : \mathrm{el}(\mathrm{val}(\alpha, y))$, and consider

$$\xi =_{\mathrm{def}} (\forall y : \mathrm{el}(\alpha))(\forall x : \mathrm{el}(\mathrm{val}(\alpha, y))) \ \ (\theta \wedge \eta \ \supset \ \phi[\mathrm{val}(\gamma, \mathrm{pair}(x, y))/x]) \, .$$

We claim that $(\forall x \in \alpha)\phi \equiv \psi \equiv \xi \equiv (\forall y \in a)(\forall x \in y)\phi$ holds, which would give us the desired result. The first equivalence follows by simply unfolding the definitions, ψ implies ξ by the $\forall$-elimination rule and ξ implies ψ by the Σ-induction rule, and the third equivalence is a consequence of predicate logic rules. The proof of the claim involving the existential quantifier follows the same pattern of reasoning. $\square$

To define a separation operation we exploit essentially that we are assuming the hybrid approach to the problem of representing the notion of a small collection of elements of a type. Let $\alpha : \mathrm{Sub}(A)$ and $(x : A)\ p : \mathbb{P}$. We define

$$\{ x \in \alpha \mid p \ \} : \mathrm{Sub}(A)$$

as the small subclass γ of A such that $\mathrm{el}(\gamma) = \mathrm{el}(\alpha) : \mathrm{type}$ and, for $x : \mathrm{el}(\alpha)$, the judgements $\mathrm{dom}(\gamma, x) \equiv \mathrm{dom}(\alpha, x) \wedge \tau(p)$ and $\mathrm{val}(\gamma, x) = \mathrm{val}(\alpha, x) : A$ are derivable.

Lemma 3.5. *Let $\alpha : \mathrm{Sub}(A)$ and $(x : A)\ p : \mathbb{P}$. There exists $\gamma : \mathrm{Sub}(A)$ such that the judgements*

$$\begin{aligned} (\forall x \in \gamma)\phi \quad &\equiv \quad (\forall x \in \alpha)(\tau(p) \supset \phi) \\ (\exists x \in \gamma)\phi \quad &\equiv \quad (\exists x \in \alpha)(\tau(p) \wedge \phi) \end{aligned}$$

are derivable.

Proof. If $\gamma = \{x \in \alpha \mid p\} : \mathrm{Sub}(A)$ then it holds that

$$(\forall x \in \gamma)\phi \equiv (\forall x : \mathrm{el}(\alpha))\big(\big(\,\mathrm{dom}(\alpha, x) \wedge \tau\, p[\mathrm{val}(\alpha, x)/x]\big) \supset \phi[\mathrm{val}(\alpha, x)/x]\big)$$

The rules of predicate logic imply that the right-hand side in the above equivalence is in turn equivalent to $(\forall x \in \alpha)(\tau(p) \supset \phi)$ as required. The proof of the statement involving existential quantification is analogous. $\square$

The next definition introduces a type-theoretic analog of the constructions allowed by the Replacement axiom of set theory [5]. For $A, B : \mathrm{type}$, $\alpha : \mathrm{Sub}(A)$ and $(x : A)\ b : B$, define

$$\{b \mid x \in \alpha\} : \mathrm{Sub}(B)$$

as the small subclass β of B such that $\mathrm{el}(\beta) = \mathrm{el}(\alpha) : \mathrm{type}$ and, for $x : \mathrm{el}(\alpha)$, the judgements $\mathrm{dom}(\beta, x) \equiv \mathrm{dom}(\alpha, x)$ and $\mathrm{val}(\beta, x) = b[\mathrm{val}(\alpha, x)/x] : B$ are derivable.

Lemma 3.6. *Let* $\alpha : \mathrm{Sub}(A)$, $(x : A)\ b : B$ *and* $(y : B)\ \psi : \mathrm{prop}$. *There exists* $\beta : \mathrm{Sub}(B)$ *such that the judgements*

$$\begin{aligned} (\forall y \in \beta)\psi &\equiv (\forall x \in \alpha)\psi[b/y] \\ (\exists y \in \beta)\psi &\equiv (\exists x \in \alpha)\psi[b/y] \end{aligned}$$

are derivable.

Proof. First of all, observe that we can assume that x is not a free variable in ψ. Let $\beta =_{\mathrm{def}} \{b \mid x \in \alpha\} : \mathrm{Sub}(B)$, By unfolding definitions and performing the appropriate substitutions we can derive that $(\forall y \in \beta)\psi$ is logically equivalent to

$$(\forall y : \mathrm{el}(\alpha))(\mathrm{dom}(\alpha, y) \supset \psi[b[\mathrm{val}(\alpha, y)/x]/y])$$

and this is in turn equivalent to $(\forall x \in \alpha)\psi[b/y]$, since we assumed that x is not a free variable in ψ. The statement involving existential quantification can be proved in a similar way. □

We conclude this series of lemmas by transferring to logic-enriched type theories a familiar fact of constructive set theories: the correspondence between the class of subsets of a singleton set and restricted sentences, as discussed in [5] and [11, Section 2.3]. In logic-enriched type theories the role of the class of all subsets of a singleton set is played by the type of small subclasses of the type $\mathbb{1}$ and the role of restricted sentences is played by elements of $\mathbb{P}$. Define

$$\mathrm{ext}(p) : \mathrm{Sub}(\mathbb{1})$$

as the small subclass γ of $\mathbb{1}$ such that $\mathrm{el}(\gamma) = \widehat{\mathbb{1}} : \mathrm{type}$ and, for $x : \mathbb{1}$, the judgements $\mathrm{dom}(\gamma, x) \equiv \tau(p)$ and $\mathrm{val}(\gamma, x) = x : \mathbb{1}$ are derivable.

Lemma 3.7. *Let* $p : \mathbb{P}$ *and* $\psi : \mathrm{prop}$. *There is* $\gamma : \mathrm{Sub}(\mathbb{1})$ *such that the judgements*

$$\begin{aligned} (\forall_{\_} \in \gamma)\ \psi &\equiv \tau(p) \supset \psi \\ (\exists_{\_} \in \gamma)\ \psi &\equiv \tau(p) \wedge \psi \end{aligned}$$

are derivable.

Proof. Let $\gamma = \mathrm{ext}(p) : \mathrm{Sub}(\mathbb{1})$. The conclusion then follows by unfolding the definitions. □

We now formulate the collection rules for small subclasses. Given $A : \mathrm{type}$, $B : \mathrm{type}$, and $(x : A, y : B)\ \phi : \mathrm{prop}$, for $\alpha : \mathrm{Sub}(A)$ and $\beta : \mathrm{Sub}(B)$ we define

$$(\forall\exists\ \tfrac{x \in \alpha}{y \in \beta})\ \phi \ =_{\mathrm{def}} (\forall x \in \alpha)\,(\exists y \in \beta)\phi \ \wedge\ (\forall y \in \beta)\,(\exists x \in \alpha)\phi \ : \mathrm{prop}$$

The Strong Collection rule is

$$\frac{A, B : \mathrm{type} \qquad \alpha : \mathrm{Sub}(A) \qquad (x : A, y : B)\ \phi : \mathrm{prop}}{(\forall x \in \alpha)(\exists y : B)\phi \ \Rightarrow\ (\exists v : \mathrm{Sub}(B))(\forall\exists\ \tfrac{x \in \alpha}{y \in v})\ \phi}$$

and the Subset Collection rule is

$$\frac{A, B, C : \mathrm{type} \quad \alpha : \mathrm{Sub}(A) \quad \beta : \mathrm{Sub}(B) \quad (x : A, y : B, z : C)\ \phi : \mathrm{prop}}{(\exists u : \mathrm{Sub}^2(B))(\forall z : C)\big((\forall x \in \alpha)(\exists y \in \beta)\phi \supset (\exists v \in u)(\forall \exists\ {}^{x \in \alpha}_{y \in v})\ \phi\big)}$$

We will show in Section 4 that these rules are derivable under the propositions-as-types interpretation of logic. We write ML(COLL) for the extension of the logic-enriched type theory $\mathrm{ML}_1 + \mathrm{W} + \mathrm{IL}_1 + \mathrm{IND}$ obtained by adding the Strong Collection and the Subset Collection rules. Recalling that CZF^- is the subsystem of CZF obtained from CZF by leaving out the Subset Collection axiom scheme, and that in CZF^- it is not possible to derive the Exponentiation axiom, asserting that the class of functions from a set to a set is again a set [18], it is natural to define $\mathrm{ML}(\mathrm{COLL}^-)$ as the type theory obtained from ML(COLL) by omitting the Subset Collection rule and the rules reflecting Π-types in the type universe.

Iterative small classes. Let us begin by defining the type

$$\mathbb{V} =_{\mathrm{def}} \big(\mathbb{W}z : (\Sigma x : \mathbb{U})(\mathbb{T}(x) \to \mathbb{P})\big)\, \mathbb{T}(z.1)$$

A canonical element of $\mathbb{V}$ has form $\mathrm{sup}(\mathrm{pair}(a,p), f) : \mathbb{V}$ where $a : \mathbb{U}$, $p : \mathbb{T}(a) \to \mathbb{P}$ and $f : \mathbb{T}(a) \to \mathbb{V}$. Such an object can be thought of as the 'set of' $\mathrm{app}(f, x) : \mathbb{V}$ for $x : \mathbb{T}(a)$ such that $\tau(\mathrm{app}(p, x))$ holds. We will refer to the elements of this type as iterative small classes. As usual, it is convenient to introduce some explicitly defined expressions. We define, for $\alpha : \mathrm{Sub}(\mathbb{V})$

$$\mathrm{set}(\alpha) =_{\mathrm{def}} \mathrm{sup}(\mathrm{pair}(\alpha.1, \alpha.2.1), \alpha.2.2) : \mathbb{V}$$

and, for $\alpha : \mathbb{V}$

$$\mathrm{sub}(\alpha) =_{\mathrm{def}} \mathrm{r_W}(\alpha, (u, v)\, \mathrm{pair}(u.1, \mathrm{pair}(u.2, v)) : \mathrm{Sub}(\mathbb{V})$$

Observe that there is a correspondence between elements of $\mathbb{V}$ and elements of $\mathrm{Sub}(\mathbb{V})$. The next lemma shows a first property of this correspondence.

Lemma 3.8. *For $a : \mathbb{U}$, $p : \mathbb{T}(a) \to \mathbb{P}$, and $f : \mathbb{T}(a) \to \mathbb{V}$ the judgements*

$$\mathrm{set}(\mathrm{pair}(a, \mathrm{pair}(p, f))) = \mathrm{sup}(\mathrm{pair}(a, p), f) : \mathbb{V}$$
$$\mathrm{sub}(\mathrm{sup}(\mathrm{pair}(a, p), f) = \mathrm{pair}(a, \mathrm{pair}(p, f)) : \mathrm{Sub}(\mathbb{V})$$

are derivable.

Proof. The judgements follow from the computation rules for W-types and Σ-types. □

Lemma 3.9. *For $(x : \mathbb{V})\ \phi : \mathrm{prop}$ and $(y : \mathrm{Sub}\,\mathbb{V})\ \psi : \mathrm{prop}$ the judgements*

$$(\forall x : \mathbb{V})\phi \equiv (\forall y : \mathrm{Sub}(\mathbb{V}))\phi[\mathrm{set}(y)/x]$$
$$(\exists x : \mathbb{V})\phi \equiv (\exists y : \mathrm{Sub}(\mathbb{V}))\phi[\mathrm{set}(y)/x]$$

are derivable.

Proof. To prove the judgement

$$(\forall x : \mathbb{V})\phi \equiv (\forall y : \mathrm{Sub}(\mathbb{V}))\phi[\mathrm{set}(y)/x]$$

we consider

$$\psi =_{\mathrm{def}} (\forall x : \mathbb{U})(\forall y : \mathbb{T}(x) \to \mathbb{P})(\forall z : \mathbb{T}(x) \to \mathbb{V})\phi[\sup(\mathrm{pair}(x,y),z)/x]$$

We show that

$$(\forall x : \mathbb{V})\phi \equiv \psi \equiv (\forall y : \mathrm{Sub}(\mathbb{V}))\phi[\mathrm{set}(y)/x] \tag{24}$$

The first equivalence in (24) can be proved as follows: the left-to-right implication is proved with the $\forall$-elimination rule, while the right-to-left is proved by W-induction. The second equivalence in (24) can instead be obtained as we describe now. The left-to-right implication is a consequence of the Σ-induction rule, while the right-to-left implication follows by the $\forall$-elimination rule and Lemma 3.8. □

We introduce versions of quantification over iterative small classes using Lemma 3.8 and the definitions expressing quantification over a small subclass of a type in (17) and (18). For $\alpha : \mathbb{V}$, and $(x : \mathbb{V})\,\phi : \mathrm{prop}$ we define

$$\begin{aligned}
(\forall x \in \alpha)\phi \quad &=_{\mathrm{def}} \quad (\forall x \in \mathrm{sub}(\alpha))\phi : \mathrm{prop} \\
(\exists x \in \alpha)\phi \quad &=_{\mathrm{def}} \quad (\exists x \in \mathrm{sub}(\alpha))\phi : \mathrm{prop}
\end{aligned}$$

and, for $(x : \mathbb{V})\ p : \mathbb{P}$, we let

$$\begin{aligned}
(\forall x \in \alpha)p \quad &=_{\mathrm{def}} \quad (\forall x \in \mathrm{sub}(\alpha))p : \mathbb{P} \\
(\exists x \in \alpha)p \quad &=_{\mathrm{def}} \quad (\exists x \in \mathrm{sub}(\alpha))p : \mathbb{P}
\end{aligned}$$

Again, these definitions can easily be shown to be compatible.

We now want to apply a special instance of the elimination rule for the W-type $\mathbb{V}$ of iterative small classes and define a type-theoretic counterpart to the set-theoretic extensional equality relation. To do so, let us introduce some auxiliary definitions. Define $A =_{\mathrm{def}} (\Sigma x : \mathbb{U})(\mathbb{T}(x) \to \mathbb{P})$ and, for $u : A$, let

$$\begin{aligned}
B \quad &=_{\mathrm{def}} \quad \mathbb{T}(u.1) \to \mathbb{P} \\
C \quad &=_{\mathrm{def}} \quad \mathbb{T}(u.1) \to (\mathbb{V} \to \mathbb{P})
\end{aligned}$$

and finally $D =_{\mathrm{def}} \mathbb{V} \to \mathbb{P}$. For $u : A$, $z : B$, and $w : C$ we can then define

$$g =_{\mathrm{def}} (\lambda v : \mathbb{V})(g_1 \wedge g_2) : D$$

where, for $v : \mathbb{V}$ we let

$$\begin{aligned}
g_1 \quad &=_{\mathrm{def}} \quad (\forall x : u.1)\,\mathrm{app}(u.2,x) \supset (\exists y \in v)\,\mathrm{app}(\mathrm{app}(w,x),y) : \mathbb{P} \\
g_2 \quad &=_{\mathrm{def}} \quad (\forall y \in v)(\exists x : u.1)\,\mathrm{app}(u.2,x) \wedge \mathrm{app}(\mathrm{app}(w,x),y) : \mathbb{P}
\end{aligned}$$

We can then apply the following instance of the elimination rule for $\mathbb{V}$

$$\frac{D : \mathrm{type} \quad (u : A, z : B, w : C)\ g : D \quad \alpha : \mathbb{V}}{\mathrm{r_W}(\alpha, (u,z,w)d) : D}$$

and define

$$\alpha \approx \beta =_{\mathrm{def}} \mathrm{app}(\mathrm{r_W}(\alpha, (u, z, w)g), \beta) : \mathbb{P} \tag{25}$$

for $\alpha, \beta : \mathbb{V}$.

Lemma 3.10. *For $\alpha, \beta : V$ the judgement*

$$\alpha \approx \beta \quad \equiv \quad (\forall\exists\ \tfrac{x \in \alpha}{y \in \beta})\ x \approx y$$

is derivable.

Proof. The proof involves unfolding the appropriate definitions and applying the computation rule for the W-type $\mathbb{V}$ of iterative small classes. ☐

Sets-as-trees. We now define the generalized type-theoretic interpretation of CZF. Recall from the introduction that we are assuming that CZF is formulated in a language $\mathcal{L}$ with equality and primitive restricted quantifiers, in which the membership relation is defined. Let us also assume that the symbols for variables for sets of CZF coincide with the symbols for variables of type $\mathbb{V}$. We define two interpretations. The first is indicated with $[\![\cdot]\!]$ and applies to arbitrary formulas. The second is indicated with $(\!|\cdot|\!)$ and applies only to restricted formulas. Both interpretations are defined in Table 3, where $\star$ is $\wedge$, $\vee$ or $\supset$, and ∇ is $\forall$ or $\exists$.

$$\begin{array}{rlrl}
[\![x = y]\!] & =_{\mathrm{def}} x \approx y & (\!|x = y|\!) & =_{\mathrm{def}} x \approx y \\
[\![\phi_1 \star \phi_2]\!] & =_{\mathrm{def}} [\![\phi_1]\!] \star [\![\phi_2]\!] & (\!|\theta_1 \star \theta_2|\!) & =_{\mathrm{def}} (\!|\theta_1|\!) \star (\!|\theta_2|\!) \\
[\![(\nabla x \in \alpha)\, \phi]\!] & =_{\mathrm{def}} (\nabla x \in \alpha)[\![\phi]\!] & (\!|(\nabla x \in \alpha)\theta|\!) & =_{\mathrm{def}} (\nabla x \in \alpha)(\!|\theta|\!) \\
[\![(\nabla x)\phi]\!] & =_{\mathrm{def}} (\nabla x : \mathbb{V})[\![\phi]\!] & &
\end{array}$$

TABLE 3. Interpretation of the language of CZF.

Lemma 3.11. *Let θ, ϕ be formulas of $\mathcal{L}$ with free variables $x_1, \ldots, x_n$, and assume that θ is restricted. Then the judgements*

$$\begin{array}{ll}
(x_1 : \mathbb{V}, \ldots, x_n : \mathbb{V}) & [\![\phi]\!] : \mathrm{prop} \\
(x_1 : \mathbb{V}, \ldots, x_n : \mathbb{V}) & (\!|\theta|\!) : \mathbb{P} \\
(x_1 : \mathbb{V}, \ldots, x_n : \mathbb{V}) & \tau(\!|\theta|\!) \equiv [\![\theta]\!]
\end{array}$$

are derivable.

Proof. Reasoning by structural induction suffices to prove the claim. ☐

Definition 3.12. A formula ϕ of $\mathcal{L}$ with free variables $x_1, \ldots, x_n$ will be said to be *valid* if the judgement

$$(x_1 : \mathbb{V}, \ldots, x_n : \mathbb{V}) \quad [\![\phi]\!]$$

is derivable. We say that the generalized type-theoretic interpretation of CZF is valid if each axiom scheme is valid.

We begin by establishing that the logical axioms of CZF are valid.

Lemma 3.13. *Let* $(x : \mathbb{V})\ \phi : \mathrm{prop}$ *and* $\alpha, \beta : \mathbb{V}$. *Then the judgement*

$$[\![\phi[\alpha/x]]\!] \wedge [\![\alpha = \beta]\!] \supset [\![\phi[\beta/x]]\!]$$

is derivable.

Proof. The claim follows by structural induction on ϕ. □

Given Lemma 3.13, the predicate logic rules imply that all the logical axioms for CZF, and in particular the ones regarding restricted quantifiers, are valid. The next lemma takes care of the structural axioms of CZF.

Lemma 3.14. *Extensionality and Set Induction are valid.*

Proof. Validity of Extensionality follows from Lemma 3.10. Validity of Set Induction is a consequence of the induction rule for the W-type $\mathbb{V}$ of iterative small classes. □

Lemma 3.15. *Pairing, Union, Infinity and Restricted Separation are valid.*

Proof. Lemma 3.2 helps us to prove validity of Pairing. Let $\alpha, \beta : \mathbb{V}$ and define $\gamma =_{\mathrm{def}} \mathrm{set}(\{\alpha, \beta\}) : \mathbb{V}$. By Lemma 3.2 we have

$$\begin{array}{rcl}(\forall x \in \gamma)(x \approx \alpha \vee x \approx \beta) & \equiv & (\forall x \in \{\alpha, \beta\})(x \approx \alpha \vee x \approx \beta) \\ & \equiv & (\alpha \approx \alpha \vee \alpha \approx \beta) \wedge (\beta \approx \alpha \vee \beta \approx \beta)\end{array}$$

Similarly we get

$$\begin{array}{rcl}\alpha \in \gamma \wedge \beta \in \gamma & \equiv & (\exists x \in \{\alpha, \beta\})(x \approx \alpha) \wedge (\exists x \in \{\alpha, \beta\})(x \approx \beta) \\ & \equiv & (\alpha \approx \alpha \vee \alpha \approx \beta) \wedge (\beta \approx \alpha \vee \beta \approx \beta)\end{array}$$

as required. Soundness of Union follows in a similar way from Lemma 3.4. For Infinity we exploit Lemma 3.1 and Corollary 3.3. We let $\omega : \mathrm{Sub}(\mathbb{V})$ be the small subclass of $\mathbb{V}$ such that $\mathrm{el}(\omega) = \mathbb{N} : \mathrm{type}$, and for $n : \mathbb{N}$, the judgements $\mathrm{dom}(\omega, n) \equiv \top$ and

$$\mathrm{val}(\omega, n) = \mathrm{r}_{\mathbb{N}}(n, \mathrm{set}(\emptyset_{\mathbb{V}}), (x, y)\,\mathrm{set}(\{y\})) : \mathbb{V}$$

are derivable, where we used the elimination rule for the type $\mathbb{N}$ given in Appendix A. It is immediate to see that $\mathrm{set}(\omega) : \mathbb{V}$ can be used to show the soundness of Infinity. Validity of Restricted Separation follows from Lemma 3.5. □

The collection rules for small subclasses that are part of ML(COLL) allow us to prove the validity of all the instances of the collection axiom schemes of CZF.

Lemma 3.16. *Strong Collection and Subset Collection are valid.*

Proof. The interpretation of each instance of these axiom schemes can be proved from a suitable instance of the corresponding type-theoretic rule using the correspondence between $\mathbb{V}$ and $\mathrm{Sub}(\mathbb{V})$. □

These results provide a proof of our first main result.

Theorem 3.17. *The logic-enriched type theory* ML(COLL) *proves that the generalised type-theoretic interpretation of* CZF *in* $(\mathbb{V}, \approx)$ *is valid.*

4. An analysis of the original type-theoretic interpretation

The original interpretation of CZF in the type theory $\mathrm{ML}_1 + \mathrm{W}$ can be viewed as taking place in two steps: an interpretation of CZF in the logic-enriched type theory $\mathrm{ML}(\mathrm{AC} + \mathrm{PU})$ followed by the propositions-as-types interpretation of $\mathrm{ML}(\mathrm{AC} + \mathrm{PU})$ in $\mathrm{ML}_1 + \mathrm{W}$. The generalised interpretation of CZF presented in Section 3 is concerned with a strengthening of the interpretation of CZF into $\mathrm{ML}(\mathrm{AC} + \mathrm{PU})$ so as to interpret CZF in the logic-enriched type theory ML(COLL) obtained from $\mathrm{ML}_1 + \mathrm{W} + \mathrm{IL}_1 + \mathrm{IND}$ by adding two type-theoretic collection principles corresponding to the two collection principles of CZF. In this section we describe how the interpretations of CZF in ML(COLL) and in $\mathrm{ML}(\mathrm{AC} + \mathrm{PU})$ are related. In particular, we will prove that the two type-theoretic collection principles can be derived in $\mathrm{ML}(\mathrm{AC} + \mathrm{PU})$ so that the new interpretation is indeed a refinement of the old one.

The interpretation of CZF in the logic-enriched type theory $\mathrm{ML}(\mathrm{AC} + \mathrm{PU})$ assumes the combinatorial notion of small collection, as discussed in Section 3, to interpret the notion of 'set of'. We will therefore refer to this interpretation as the combinatorial intepretation of CZF. The combinatorial approach leads naturally to the use of the type $\mathrm{V} =_{\mathrm{def}} (\mathbb{W}x : \mathbb{U})\, \mathbb{T}(x)$ to interpret the universe of sets of CZF. Indeed, for each small collection $\mathrm{pair}(a, f) : \mathrm{Fam}(\mathrm{V})$, there is $\sup(a, f) : \mathrm{V}$. Each formula ϕ of CZF with free variables $x_1, \ldots, x_n$ is then interpreted as $[\![\phi]\!]$ such that the judgement

$$(x_1 : \mathrm{V}, \ldots, x_n : \mathrm{V})\ [\![\phi]\!] : \mathrm{prop}$$

can be derived, and each restricted formula θ with free variables $x_1, \ldots, x_n$ is interpreted as $(\!|\theta|\!)$ such that the judgements

$$(x_1 : \mathrm{V}, \ldots, x_n : \mathrm{V})\ (\!|\theta|\!) : \mathbb{P}$$

and

$$(x_1 : \mathrm{V}, \ldots, x_n : \mathrm{V})\ \tau(\!|\theta|\!) \equiv [\![\theta]\!]$$

can be derived. The validity of the axioms of CZF under the original type-theoretic interpretation relies on the properties of the types of small families of objects. We isolate the relevant properties below, but we prefer to avoid giving detailed proofs, since we have presented detailed proofs of the corresponding statements for small subclasses in Section 3.

Small families. The next lemma states that small families support the definition of some basic set-theoretical constructs: the empty set, pairing, and union.

Lemma 4.1. *Let* $A : \mathrm{type}$ *and* $(x : A)\ \phi : \mathrm{prop}$.

(*i*) *There exists* $\gamma : \mathrm{Fam}(A)$ *such that the judgements*

$$\begin{aligned}(\forall x \in \gamma)\ \phi &\equiv \top\\ (\exists x \in \gamma)\ \phi &\equiv \bot\end{aligned}$$

are derivable.

(*ii*) *Let* $a_1, a_2 : A$. *There exists* $\gamma : \mathrm{Fam}(A)$ *such that the judgements*

$$\begin{aligned}(\forall x \in \gamma)\ \phi &\equiv \phi[a_1/x] \wedge \phi[a_2/x]\\ (\exists x \in \gamma)\ \phi &\equiv \phi[a_1/x] \vee \phi[a_2/x]\end{aligned}$$

are derivable.

(*iii*) *Let* $\alpha : \mathrm{Fam}^2(A)$, *where* $\mathrm{Fam}^2(A) = \mathrm{Fam}(\mathrm{Fam}(A))$. *There exists* $\gamma : \mathrm{Fam}(A)$ *such that the judgements*

$$\begin{aligned}(\forall x \in \gamma)\ \phi &\equiv (\forall y \in \alpha)(\forall x \in y)\ \phi\\ (\exists x \in \gamma)\ \phi &\equiv (\exists y \in \alpha)(\exists x \in y)\ \phi\end{aligned}$$

are derivable.

Proof. The proofs of the three parts of the Lemma are analogous to the ones of Lemma 3.1, Lemma 3.2, and Lemma 3.4 respectively. □

The previous lemma states the key facts needed to obtain the validity of Pairing and Union in the original type-theoretic interpretation. As we discussed in Section 3 it is necessary to assume the rule (PU) in order to obtain a version of Restricted Separation when the combinatorial notion of small collection is assumed.

Lemma 4.2. *Let* $A : \mathrm{type}$, $\alpha : \mathrm{Fam}(A)$ *and* $(x : A)\ p : \mathbb{P}$. *Assuming* (PU) *there exists* $\gamma : \mathrm{Fam}(A)$ *such that, for* $(x : A)\ \phi : \mathrm{prop}$, *the judgements*

$$\begin{aligned}(\forall x \in \gamma)\ \phi &\equiv (\forall x \in \alpha)(\tau(p) \supset \phi)\\ (\exists x \in \gamma)\ \phi &\equiv (\exists x \in \alpha)(\tau(p) \wedge \phi)\end{aligned}$$

can be derived.

Proof. The definition of the appropriate $\gamma : \mathrm{Sub}(A)$ is given in (20) and (21). The required judgements follow by unfolding the relevant definitions. □

The type-theoretic axiom of choice is crucial to prove the validity of the two collection schemes of CZF in the original type-theoretic interpretation. Indeed, the rule (AC) allows us to prove the next lemma, from which the two collection rules concerning small families follow easily.

Lemma 4.3. *Assuming* (AC), *the rules*

$$\frac{\alpha : \mathrm{Fam}(A) \quad B : \mathrm{type} \quad (x : A, y : B)\ \phi : \mathrm{prop}}{(\forall x \in \alpha)(\exists y : B) \Rightarrow (\exists f : \mathrm{el}(\alpha) \to B)(\forall x \in \alpha)\phi[\mathrm{app}(f,x)/y]}$$

and

$$\frac{\alpha : \mathrm{Fam}(A) \quad \beta : \mathrm{Fam}(B) \quad (x : A, y : B)\ \phi : \mathrm{prop}}{(\forall x \in \alpha)(\exists y \in \beta)\phi \Rightarrow (\exists f : \mathrm{el}(\alpha) \to \mathrm{el}(\beta))(\forall x \in \alpha)\phi[\mathrm{val}(\beta, \mathrm{app}(f,x))/y]}$$

are derivable.

For types A and B, $\alpha : \mathrm{Fam}(A)$ and $\beta : \mathrm{Fam}(B)$ and $(x : A, y : B)\ \phi : \mathrm{prop}$ we define

$$(\forall\exists\ {}^{x\in\alpha}_{y\in\beta})\ \phi =_{\mathrm{def}} (\forall x \in \alpha)(\exists y \in \beta)\phi \wedge (\forall y \in \beta)(\exists x \in \alpha)\phi : \mathrm{prop}$$

We can now obtain the collection rules for small families. These rules are formulated just like those rules for small collections presented in Section 3.

Proposition 4.4. *Assuming* (AC), *the rules*

$$\frac{A, B : \mathrm{type} \qquad \alpha : \mathrm{Fam}(A) \qquad (x : A, y : B)\ \phi : \mathrm{prop}}{(\forall x \in \alpha)(\exists y : B)\phi \Rightarrow (\exists v : \mathrm{Fam}(B))(\forall\exists\ {}^{x\in\alpha}_{y\in v})\,\phi}$$

and

$$\frac{A, B, C : \mathrm{type} \quad \alpha : \mathrm{Fam}(A) \quad \beta : \mathrm{Fam}(B) \quad (x : A, y : B, z : C)\ \psi : \mathrm{prop}}{(\exists u : \mathrm{Fam}^2(B))(\forall z : C)\big((\forall x \in \alpha)(\exists y \in \beta)\psi \supset (\exists v \in u)(\forall\exists\ {}^{x\in\alpha}_{y\in v})\ \psi\big)}$$

are derivable.

Proof. To derive the Strong Collection rule for small families, let us assume that $(\forall x \in \alpha)\,(\exists y : B)\,\phi$ holds. By Lemma 4.3 there is $f : \mathrm{el}(\alpha) \to B$ such that $(\forall x \in \alpha)\ \phi[\mathrm{app}(f, x)/y]$ holds. Once we define

$$\beta =_{\mathrm{def}} \mathrm{pair}(\mathrm{el}(\alpha), f) : \mathrm{Fam}(B)$$

it is straightforward to show that $(\forall\exists\ {}^{x\in\alpha}_{y\in\beta})\ \phi$ holds, as required.

We now derive the Subset Collection rule for small families. Let $z : C$ and assume $(\forall x \in \alpha)(\exists y \in \beta)\phi$. An application of Lemma 4.3 shows that there is $f : \mathrm{el}(\alpha) \to \mathrm{el}(\beta)$ such that

$$(\forall x \in \mathrm{el}(\alpha))\phi[\mathrm{val}(\beta, \mathrm{app}(f, x))/y]$$

holds. Define

$$\beta' =_{\mathrm{def}} \mathrm{pair}(\mathrm{el}(\alpha), (\lambda x : \mathrm{el}(\alpha))\,\mathrm{val}(\beta, \mathrm{app}(f, x))) : \mathrm{Fam}(B)$$

We define an element $\delta : \mathrm{Fam}^2(B)$, independent of ϕ and $z : C$, such that $\mathrm{el}(\delta) = \mathrm{el}(\alpha) \to \mathrm{el}(\beta) : \mathbb{U}$ and, for $k : \mathrm{el}(\alpha) \to \mathrm{el}(\beta)$

$$\mathrm{val}(\delta, k) = \mathrm{pair}(\mathrm{el}(\alpha), (\lambda x : \mathrm{el}(\alpha))\,\mathrm{val}(\beta, \mathrm{app}(k, x)))) : \mathrm{Fam}(B)$$

Observe that $f : \mathrm{el}(\alpha) \to \mathrm{el}(\beta)$ and $\mathrm{val}(\delta, f) = \beta' : \mathrm{Fam}(B)$. We have therefore obtained $\delta : \mathrm{Fam}^2(B)$ such that

$$(\exists v \in \delta)(\forall\exists\ {}^{x\in\alpha}_{y\in v})\,\phi$$

holds, as required. □

The interpretation of CZF in ML(AC + PU) can now be developed analogously to the generalised type-theoretic interpretation of Section 3, except that we use the type $\mathrm{V} =_{\mathrm{def}} (\mathbb{W}x : \mathbb{U})\,\mathbb{T}(x)$ instead of the type $\mathbb{V}$ to interpret the universe of sets of CZF. Let us sketch the general outline of this interpretation. First, it is necessary to develop the appropriate notation to express quantification over an element of V. This involves proving counterparts of Lemma 3.8 and Lemma 3.9. Secondly, one defines an analog for the type V of the extensional equality that we defined for the type $\mathbb{V}$ in (25). For $\alpha_1, \alpha_2 : \mathrm{V}$ we write $\alpha_1 \approx \alpha_2 : \mathrm{prop}$ to denote it. Finally, one gives an interpretation of the language of CZF in ML(AC + PU) as in Table 3 except that the type V is used instead of the type $\mathbb{V}$ to interpret unrestricted quantifiers. The following result can then be proved.

Theorem 4.5. *The logic-enriched type theory* ML(AC + PU) *proves that the combinatorial interpretation of* CZF *in* $(\mathrm{V}, \approx)$ *is valid.*

Proof. Validity of Extensionality and Set Induction follows from the definition of the extensional equality on V and from the induction rule of the W-type V, respectively. Lemma 4.1 implies the validity of Pairing and Union, and Lemma 4.2 the validity of Restricted Separation. The proof of validity of Infinity is essentially as in the generalised type-theoretic interpretation. Finally, the validity of the Strong Collection and Subset Collection schemes follows from Proposition 4.4. □

Propositions-as-types interpretation of collection rules for small subclasses. We now want to relate the interpretations of CZF in ML(COLL) and in ML(AC + PU). To do so, we show that the Strong Collection and Subset Collection rules for small subclasses are justified under the propositions-as-types interpretation of logic. Let us use the function $\mathrm{t} : \mathbb{P} \to \mathbb{U}$ of Lemma 2.1 to define, for $\alpha : \mathrm{Sub}(A)$

$$\mathrm{comp}(\alpha) =_{\mathrm{def}} (\Sigma x : \mathrm{el}(\alpha))\, \mathrm{app}(\mathrm{t}, \mathrm{dom}(\alpha, x)) : \mathbb{U}$$

For $\alpha : \mathrm{Sub}(A)$ and $\sigma : \mathrm{Fam}(A)$ we can then let

$$\begin{array}{lll} i(\alpha) & =_{\mathrm{def}} & \mathrm{pair}(\mathrm{comp}(\alpha), (\lambda z : \mathrm{comp}(\alpha))\, \mathrm{val}(\alpha, z.1)) : \mathrm{Fam}(A) \\ j(\sigma) & =_{\mathrm{def}} & \mathrm{pair}(\mathrm{el}(\sigma), \mathrm{pair}((\lambda\_ : \mathrm{el}(\sigma))\top, (\lambda x : \mathrm{el}(\sigma))\, \mathrm{val}(\sigma, x))) : \mathrm{Sub}(A) \end{array}$$

The next lemma relates the definitions in (14) and (15) with the ones in (17) and (18).

Lemma 4.6. *Let* $(x : A)\ \phi : \mathrm{prop}$. *Assuming* (AC) *and* (PU), *for* $\alpha : \mathrm{Sub}(A)$ *and* $\sigma : \mathrm{Fam}(A)$ *the judgements*

$$\begin{array}{rcl} (\nabla x \in \alpha)\phi & \equiv & (\nabla x \in i(\alpha))\phi \\ (\nabla x \in \sigma)\phi & \equiv & (\nabla x \in j(\sigma))\phi \end{array}$$

where ∇ *is either* $\forall$ *or* $\exists$, *are derivable.*

Proof. Let us prove the judgements involving universal quantification. For the first part, observe that the rule (PU) implies

$$(\forall x : \mathrm{el}(\alpha))\, \mathrm{dom}(\alpha, x) \supset \phi[\mathrm{val}(\alpha, x)/x] \equiv (\forall z : \mathrm{comp}(\alpha))\phi[\mathrm{val}(\alpha, z.1)/x]$$

By unfolding the definitions it follows that $(\forall x \in \alpha)\phi \equiv (\forall x \in i(\alpha))\phi$ holds. For the second part, observe that

$$(\forall x \in \mathrm{el}(\sigma))\phi[\mathrm{val}(\sigma, x)/x] \equiv (\forall x \in j(\sigma))\phi$$

holds, and this proves the desired claim. □

Theorem 4.7. *Assuming* (AC) *and* (PU)*, the Strong Collection and Subset Collection rules are derivable.*

Proof. The claim follows from Proposition 4.4 and Lemma 4.6. □

We now have two interpretations of CZF in ML(AC + PU), one using V and the other using $\mathbb{V}$. But in fact these are essentially the same. This is because in ML(AC + PU) one can prove that there are inverse isomorphisms between the structure on V with extensional equality and extensional membership and the same structure on $\mathbb{V}$. The key advantage of ML(COLL) over ML(AC + PU) is the fact that the former allows for reinterpretations of the logic which are not available for the latter, as we discuss in Section 5.

5. Reinterpreting logic

Local operators in logic-enriched type theories. In this section we introduce the notion of a local operator on the proposition universe, and discuss the reinterpreation of logic determined by it. Before introducing local operators, however, it is convenient to fix some notation and establish a simple fact that will be useful in the development of reinterpretations of logic. For $p, q : \mathbb{P}$ we define $p \leq q =_{\text{def}} \tau(p) \supset \tau(q) : \text{prop}$. It is also convenient to define, for $p : \mathbb{P}$ and $\phi : \text{prop}$

$$p \leq \phi =_{\text{def}} \tau(p) \supset \phi\,.$$

We then have the following result.

Lemma 5.1. *Let* $B : \text{type}$, $p : \mathbb{P}$ *and* $(y : B)\ \psi : \text{prop}$ *such that*

$$p \leq (\exists y : B)\psi\,.$$

Then there is $\beta : \text{Sub}(B)$ *such that* $p \leq (\exists y \in \beta)\psi$ *and* $(\forall y \in \beta)(\tau(p) \wedge \psi)$.

Proof. Let $\alpha =_{\text{def}} \text{ext}(p) : \text{Sub}(\mathbb{1})$. Use Lemma 3.7 and then apply the Strong Collection Rule to $(\forall\_ \in \alpha)(\exists y : B)\psi$. □

Definition 5.2. We say that $j : \mathbb{P} \to \mathbb{P}$ is a *local operator* if the following judgements

(i) $p \leq jp$
(ii) $p \leq q \Rightarrow jp \leq jq$
(iii) $jp \wedge jq \leq j(p \wedge q)$
(iv) $j(jp) \leq jp$

are derivable, where $jp =_{\text{def}} \text{app}(j, p)$, for $p : \mathbb{P}$.

From now on we assume given an arbitrary local operator j. For $\phi : \text{prop}$ we define

$$J\phi =_{\text{def}} (\exists y : \mathbb{P})[\tau(jy) \wedge (\tau(y) \supset \phi)]$$

First of all, we note that J and j are extensionally equal on elements of the proposition universe. The following result can be obtained by expanding the relevant definitions.

Lemma 5.3. *For* $p : \mathbb{P}$*, the judgement* $J(\tau(p)) \equiv \tau(jp)$ *is derivable.*

We now show that the operator J inherits all the properties of the local operator j.

Lemma 5.4. *Let* $\phi : \mathrm{prop}$ *and* $p : \mathbb{P}$ *such that* $p \leq J\phi$*. Then there is* $q \in \mathbb{P}$ *such that* $q \leq \phi$ *and* $p \leq jq$*.*

Proof. As $p \leq J\phi$, we have that $p \leq (\exists y : \mathbb{P})\psi$, where $\psi =_{\mathrm{def}} (\tau(jy) \wedge y \leq \phi)$. Then, by Lemma 5.1, there is $\beta : \mathrm{Sub}(\mathbb{P})$ such that

$$p \leq (\exists y \in \beta)\psi \tag{26}$$

and

$$(\forall y \in \beta)(\tau(p) \wedge \psi). \tag{27}$$

Let $q =_{\mathrm{def}} (\exists y \in \beta)y \ : \ \mathbb{P}$. Then $q \leq \phi$, as $(\forall y \in \beta)(y \leq \phi)$, by (27). Also, by (26), $p \leq (\exists y \in \beta)\tau(jy)$. As $(\forall y \in \beta)(y \leq q)$ and j is monotone we get that $(\forall y \in \beta)(jy \leq jq)$ and hence $p \leq jq$. □

The next proposition shows that the properties of j can be lifted to J. In its proof we will apply Lemma 5.4, and thus make use of the Strong Collection rule.

Proposition 5.5. *For* $\phi, \psi : \mathrm{prop}$*, the judgements*

(*i*) $\phi \supset J\phi$
(*ii*) $\phi \supset \psi \Rightarrow J\phi \supset J\psi$
(*iii*) $J\phi \wedge J\psi \supset J(\phi \wedge \psi)$
(*iv*) $J(J\phi) \supset J\phi$

are derivable.

Proof. Direct derivations suffice to prove the first three claims. The proof of the last one uses Lemma 5.4 and the fact that j is monotone. □

We define the j-interpretation of ML(COLL$^-$) into itself determined by the local operator j. This interpretation acts solely on the logic, leaving types unchanged. We define the j-interpretation by structural induction on the raw syntax of the type theory. The interpretation on formulae is defined in Table 4, where $\star$ is either $\wedge, \vee$ or $\supset$ and ∇ is either $\forall$ or $\exists$.

$$\begin{array}{rcl}
\langle \top \rangle_j & =_{\mathrm{def}} & \top \\
\langle \bot \rangle_j & =_{\mathrm{def}} & \bot \\
\langle \phi \star \psi \rangle_j & =_{\mathrm{def}} & J\langle \phi \rangle_j \star J\langle \psi \rangle_j \\
\langle (\nabla x : A)\, \phi \rangle_j & =_{\mathrm{def}} & (\nabla x : A)\, J\langle \phi \rangle_j \\
\langle \tau(a) \rangle_j & =_{\mathrm{def}} & \tau(a)
\end{array}$$

TABLE 4. Definition of the j-interpretation of formulas.

$$
\begin{array}{rcl}
\langle A : \mathrm{type} \rangle_j & =_{\mathrm{def}} & A : \mathrm{type} \\
\langle A = A' : \mathrm{type} \rangle_j & =_{\mathrm{def}} & A = A' : \mathrm{type} \\
\langle a : A \rangle_j & =_{\mathrm{def}} & a : A \\
\langle a = a' : A \rangle_j & =_{\mathrm{def}} & a = a' : A \\
\langle \phi : \mathrm{prop} \rangle_j & =_{\mathrm{def}} & \langle \phi \rangle_j : \mathrm{prop} \\
\langle (\, \phi_1, \cdots, \phi_n \Rightarrow \phi \,) \rangle_j & =_{\mathrm{def}} & J\langle \phi_1 \rangle_j, \cdots, J\langle \phi_n \rangle_j \Rightarrow J\langle \phi \rangle_j
\end{array}
$$

TABLE 5. Definition of the j-interpretation of judgment bodies

The definition of the j-interpretation of judgement bodies follows in Table 5. Given these definitions, we let the j-interpretation of judgements to be defined as follows:

$$\langle\ (\Gamma)\ \mathcal{B}\ \rangle_j =_{\mathrm{def}} (\ \Gamma\ )\ \langle \mathcal{B} \rangle_j$$

Definition 5.6. 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 $\langle (\Gamma_1)\ \mathcal{B}_1 \rangle_j$, $\ldots$, $\langle (\Gamma_n)\ \mathcal{B}_n \rangle_j$.

Proposition 5.7. *The j-interpretation of the predicate logic and induction rules of* ML(COLL) *is sound.*

Proof. The result follows by a series of routine calculations. □

We now prove three lemmas that will lead us to a proof of the result that the j-interpretation of the Strong Collection rule is sound. For $z \in \mathrm{Sub}(\mathbb{1})$, let

$$\sigma(z) =_{\mathrm{def}} J(\exists_{\_} \in z)\top : \mathrm{prop}.$$

Lemma 5.8. *For $\phi : \mathrm{prop}$, the judgement*

$$J\phi \equiv (\exists z : \mathrm{Sub}(\mathbb{1}))\big(\sigma(z) \land (\forall_{\_} \in z)\phi\big)$$

is derivable.

Proof. This is a consequence of the definition of J and of Lemma 3.7. □

Lemma 5.9. *For $A, B : \mathrm{type}$, $(x : A, y : B)\ \phi : \mathrm{prop}$, $\alpha : \mathrm{Sub}(A)$ and $\gamma : \mathrm{Sub}^2(B)$, the judgement*

$$(1) \implies (2)$$

can be derived, where

$$(1) \quad =_{\mathrm{def}} \quad (\forall\exists\ \tfrac{x \in \alpha}{w \in \gamma})\,(\exists z : \mathrm{Sub}(\mathbb{1}))[\sigma(z) \land (\forall\exists\ \tfrac{\_ \in z}{y \in w})\,\phi]$$

and

$$(2) \quad =_{\mathrm{def}} \quad (\forall x \in \alpha) J(\exists y \in \bigcup\gamma)\phi \land (\forall y \in \bigcup\gamma)(\exists x \in \alpha)\phi.$$

Proof. Assume (1). Then

$$(\forall x \in \alpha)(\exists w \in \gamma)(\exists z : \mathrm{Sub}(\mathbb{1})[\sigma(z) \wedge (\forall \_ \in z)(\exists y \in w)\phi]$$

so that

$$(\forall x \in \alpha)(\exists z : \mathrm{Sub}(\mathbb{1}))[\sigma(z) \wedge (\forall \_ \in z)(\exists w \in \gamma)(\exists y \in w)\phi]$$

and hence, by Lemma 5.8,

$$(\forall x \in \alpha) J(\exists y \in \bigcup \gamma)\phi.$$

Also,

$$(\forall w \in \gamma)(\exists x \in \alpha)(\exists z : \mathrm{Sub}(\mathbb{1}))[\sigma(z) \wedge (\forall y \in w)(\exists \_ \in z)\phi]$$

so that $(\forall w \in \gamma)(\forall y \in w)(\exists x \in \alpha)\phi$ and hence

$$(\forall y \in \bigcup \gamma)(\exists x \in \alpha)\phi.$$

Thus we have proved (2). □

Lemma 5.10. *For* A, B : type, $(x : A, y : B)\ \phi$: prop, $\alpha : \mathrm{Sub}(A)$, *the judgement*

$$\begin{aligned}&(\forall x \in \alpha)\ J(\exists y : B)\ \phi \Rightarrow \\ &\qquad (\exists v : \mathrm{Sub}(B))\big((\forall x \in \alpha)\ J(\exists y \in v)\ \phi \ \ \wedge \ \ (\forall y \in v)(\exists x \in \alpha)\ \phi\big)\end{aligned}$$

is derivable.

Proof. Assume $(\forall x \in \alpha)\ J(\exists y : B)\ \phi$. By Lemma 5.9 it suffices to show that there is $\gamma : \mathrm{Sub}^2(B)$ such that (1) of that lemma holds. By Lemma 5.8

$$(\forall x \in \alpha)(\exists z : \mathrm{Sub}(\mathbb{1}))\ [\sigma(z) \ \wedge \ (\forall \_ \in z)(\exists y : B)\ \phi].$$

By Strong Collection

$$(\forall x \in \alpha)(\exists z : \mathrm{Sub}(\mathbb{1}))\ [\sigma(z) \ \wedge \ (\exists w : \mathrm{Sub}(B))(\forall\exists\ \tfrac{\_ \in z}{y \in w})\ \phi]$$

so that

$$(\forall x \in \alpha)(\exists w : \mathrm{Sub}(B))(\exists z : \mathrm{Sub}(\mathbb{1}))\ [\sigma(z) \ \wedge \ (\forall\exists\ \tfrac{\_ \in z}{y \in w})\ \phi].$$

By Strong Collection again there is $\gamma : \mathrm{Sub}^2(B)$ such that (1) of Lemma 5.9 holds. □

Proposition 5.11. *The j-interpretation of the Strong Collection rule is sound.*

Proof. The claim is a direct consequence of Lemma 5.10. □

As we will see, it is not possible to prove that the j-interpretation of the Subset Collection rule is sound for an arbitrary local operator j. We therefore introduce the notion of a set-presented local operator and show that if j is set-presented then the j-interpretation of the Subset Collection rule is sound. The notion of set-presented local operator is closely related to the notion of set-presented closure operator or nucleus [5, 11] and inductively generated formal topology [7].

Definition 5.12. A local operator j on $\mathbb{P}$ is said to be *set-presentable* if there exists $\rho : \mathrm{Sub}(\mathbb{P})$ such that the judgement

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

is derivable.

From now on we work assuming the Subset Collection rule.

Proposition 5.13. *Let* A, B, C : type *and* $(x : A, y : B, z : C)\ \psi$: prop. *If* $\alpha : \mathrm{Sub}^2(A)$ *and* $\beta : \mathrm{Sub}(B)$ *then there is* $\gamma : \mathrm{Sub}^2(B)$ *such that the judgement*

$$(\forall w \in \alpha)(\forall z : C)\big((\forall x \in w)(\exists y \in \beta)\ \psi \quad \supset \quad (\exists v \in \gamma)(\forall\exists\ {\textstyle\frac{x \in w}{y \in v}})\ \psi\big)$$

is derivable.

Proof. For $w : \mathrm{Sub}(A), u : \mathrm{Sub}^2(B), z : C$ let

$$\theta(w, u, z) =_{\mathrm{def}} [(\forall x \in w)(\exists y \in \beta)\ \psi \quad \supset \quad (\exists v \in u)(\forall\exists\ {\textstyle\frac{x \in w}{y \in v}})\ \psi].$$

By Subset Collection

$$(\forall w \in \alpha)(\exists u : \mathrm{Sub}^2(B))(\forall z : C)\ \theta(w, u, z).$$

By Strong Collection there is $\delta : \mathrm{Sub}(\mathrm{Sub}^2(B))$ such that

$$(\forall w \in \alpha)(\exists u \in \delta)(\forall z : C)\ \theta(w, u, z).$$

Let $\gamma =_{\mathrm{def}} \bigcup \delta : \mathrm{Sub}^2(B)$. Then

$$(\forall w \in \alpha)(\forall z : C)\theta(w, \gamma, z)$$

as desired. □

The next lemma is a consequence of Lemma 3.6 and Lemma 3.7.

Lemma 5.14. *If j is a set-presentable local operator, then there exists* κ : $\mathrm{Sub}^2(\mathbb{1})$ *such that, for* ϕ : prop *the judgements*

$$J\phi \quad \equiv \quad (\exists u \in \kappa)(\forall_{-} \in u)\ \phi$$

and

$$(\forall u \in \kappa)\ \sigma(u)$$

are derivable.

Let us now assume that the local operator j is set-presentable and that $\kappa : \mathrm{Sub}^2(\mathbb{1})$ satisfies the properties of Lemma 5.14.

Lemma 5.15. *Let* A, B, C : type *and* $(x : A, y : B, z : C)\ \phi$: prop. *If* $\alpha : \mathrm{Sub}(A)$ *and* $\beta : \mathrm{Sub}(B)$ *then there is* $\gamma : \mathrm{Sub}^2(B)$ *such that the judgement*

$$\begin{aligned}(\forall z : C)\ \big\{(\forall x \in \alpha)J(\exists y \in \beta)\ \phi \quad \supset \\ (\exists v \in \gamma)[(\forall x \in \alpha)J(\exists y \in v)\ \phi \quad \wedge \quad (\forall y \in v)(\exists x \in \alpha)\ \phi]\big\}\end{aligned}$$

is derivable.

Proof. Let $\kappa : \mathrm{Sub}^2(\mathbb{1})$ satisfy the properties of Lemma 5.14. For $y : B, z' : A \times C$ let

$$\phi' \ =_{\mathrm{def}} \ \phi[z'.1, z'.2/x, z].$$

By Proposition 5.13 there is $\gamma_0 : \mathrm{Sub}^2(B)$ such that

$$(\forall u \in \kappa)(\forall z' : A \times C)\ [(\forall \_ \in u)(\exists y \in \beta)\ \phi' \ \supset \ (\exists v \in \gamma_0)(\forall\exists\ \tfrac{\_ \in u}{y \in v})\ \phi']$$

It follows that

$$(\forall u \in \kappa)(\forall x : A)(\forall z : C)\ [(\forall \_ \in u)(\exists y \in \beta)\ \phi \ \supset \ (\exists v \in \gamma_0)(\forall\exists\ \tfrac{\_ \in u}{y \in v})\ \phi].$$

For $x : A, z : C, v : \mathrm{Sub}(B)$ let

$$\psi \ =_{\mathrm{def}} \ (\exists u : \mathrm{Sub}(\mathbb{1}))\ [\sigma(u) \ \wedge \ (\forall\exists\ \tfrac{\_ \in u}{y \in v})\ \phi.]$$

By Subset Collection there is $\delta : \mathrm{Sub}(\mathrm{Sub}^2(B))$ such that

$$(\forall z : C)\ [(\forall x \in \alpha)(\exists v \in \gamma_0)\ \psi \ \supset \ (\exists w \in \delta)(\forall\exists\ \tfrac{x \in \alpha}{v \in w})\ \psi].$$

Let $\gamma \ =_{\mathrm{def}} \ \{\bigcup w \mid w \in \delta\} : \mathrm{Sub}^2(B)$. To complete the proof of the lemma let $z : C$ such that $(\forall x \in \alpha) J (\exists y \in \beta)\ \phi$. By Lemma 5.14

$$(\forall x \in \alpha)(\exists u \in \kappa)(\forall \_ \in u)(\exists y \in \beta)\ \phi.$$

so that

$$(\forall x \in \alpha)(\exists u \in \kappa)(\exists v \in \gamma_0)(\forall\exists\ \tfrac{\_ \in u}{y \in v})\ \phi.$$

As $(\forall u \in \kappa)\ \sigma(u)$

$$(\forall x \in \alpha)(\exists v \in \gamma_0)\ \psi$$

so that

$$(\exists w \in \delta)(\forall\exists\ \tfrac{x \in \alpha}{v \in w})\ \psi$$

and hence, by Lemma 5.9,

$$(\exists w \in \delta)[(\forall x \in \alpha) J (\exists y \in \bigcup w)\ \phi \ \wedge \ (\forall y \in \bigcup w)(\exists x \in \alpha)\ \phi].$$

It follows that

$$(\exists v \in \gamma)[(\forall x \in \alpha) J (\exists y \in v)\ \phi \ \wedge \ (\forall y \in v)(\exists x \in \alpha)\ \phi].$$

□

Observe that the soundness of the j-interpretation of the Subset Collection rule follows directly from the previous lemma. We can then summarise the results obtained in this section in the next theorem, that is our second main result.

Theorem 5.16. *Let j be a local operator.*

(i) The j-interpretation of each rule of $\mathrm{ML}(\mathrm{COLL}^-)$ *is sound.*

(ii) Assuming the Subset Collection rule of $\mathrm{ML}(\mathrm{COLL})$*, if j is set-presentable, then the j-interpretation of the Subset Collection rule is sound.*

Double-negation interpretation. As an application of the results just obtained we present a type-theoretic version of the double-negation interpretation. We define the double-negation local operator as follows:

$$(x : \mathbb{P})\ jx =_{\mathrm{def}} \neg\neg x : \mathbb{P}$$

where

$$\neg x =_{\mathrm{def}} x \supset \bot : \mathbb{P}$$

for $x : \mathbb{P}$. It is easy to prove that j is a local operator.

Let us point out that the operator J determined by the double negation local operator need not to be logically equivalent to double negation. In fact, for ϕ : prop it holds

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

where $\neg\phi =_{\mathrm{def}} \phi \supset \bot$, for ϕ : prop. In general it will hold only that $J\phi$ implies $\neg\neg\phi$ but not viceversa. This observation seems to isolate the main reason for which it is possible to prove the soundness of the j-interpretation of the Strong Collection rule. Since [8] it is well-known that there is a standard double-negation translation of classical Zermelo-Frankel set theory, ZF, into its intuitionistic counterpart, IZF. A close inspection of the proofs reveals however that the derivation of the standard double-negation interpretation of Collection makes use not only of the Collection axiom but also of the Full Separation axiom scheme of IZF, that is not available in generalised predicative systems like CZF or ML(COLL). This use of Full Separation seems essential if one is working with the standard double-negation translation [10]. Our definition of a variant of the double-negation translation overcomes this problem. These observations arose first in connection with the development of Heyting-valued interpretations for CZF [12]. The double-negation nucleus on the Heyting algebra of truth values corresponds indeed to a double-negation interpretation [8, 14].

Since the j-interpretation of a proposition is logically equivalent to its double-negation only for small propositions, it is natural to consider the following Restricted Excluded Middle principle (REM),

$$(x : \mathbb{P})\ \ \tau(x)\ \vee\ \neg\, \tau(x) \tag{REM}$$

and we can also consider a type-theoretic principle asserting that the double-negation local operator is set-presentable

$$(\exists r : \mathrm{Sub}(\mathbb{P}))(\forall p : \mathbb{P})\ \neg\neg\, \tau(p) \equiv (\exists q \in r)\, q \supset \tau(p)\,. \tag{DNSP}$$

Theorem 5.17.

(i) The double-negation interpretation of ML(COLL$^-$) + REM *is sound.*
(ii) Assuming (DNSP)*, the double-negation interpretation of* ML(COLL)+ REM *is sound.*

Proof. The claims are direct consequences of Theorem 5.16. □

We conclude the paper with a lower bound for the proof-theoretic strength of the logic-enriched type theory ML(COLL) + DNSP.

Corollary 5.18. *Second-order arithmetic is proof-theoretically reducible to the logic-enriched type theory* ML(COLL) + DNSP.

Proof. Consider the set theory CZF + REM that is obtained from CZF by adding a scheme asserting the law of exclued middle for restricted formulas. Using the generalised type-theoretic interpretation, this semi-classical set theory is interpretable in the logic-enriched type theory ML(COLL) + REM, which can in turn be interpreted in ML(COLL) + DNSP via the double-negation translation. The set theory CZF + REM has proof-theoretic strength at least above that of Bounded Zermelo set theory, which is obtained from Zermelo set theory by replacing the Full Separation axiom scheme with its restricted counterpart. This is because the Power Set axiom is derivable in CZF + REM, and Bounded Zermelo set theory has a double-negation interpretation into its intuitionistic counterpart, which is a subsystem of CZF + REM. □

Appendix A. Type-theoretic rules

The pure type theory ML. We present the rules of the pure type theory ML, discussed in Section 1. Informal explanations for the rules of this type theory can be found in [21].

Assumption rule. The following rule applies under the side-condition that $x \notin \mathrm{FV}(\Gamma) \cup \mathrm{FV}(\Delta)$.

$$\frac{(\Gamma, \Delta)\ \mathcal{B} \qquad A : \mathrm{type}}{(\Gamma, x : A, \Delta)\ \mathcal{B}}$$

Equality rules. From now on we suppress mention of a context that is common to both the premisses and the conclusion of a rule.

$$\frac{A : \mathrm{type}}{A = A : \mathrm{type}} \qquad \frac{A_1 = A_2 : \mathrm{type}}{A_2 = A_1 : \mathrm{type}} \qquad \frac{A_1 = A_2 : \mathrm{type} \qquad A_2 = A_3 : \mathrm{type}}{A_1 = A_3 : \mathrm{type}}$$

$$\frac{a : A}{a = a : A} \qquad \frac{a_1 = a_2 : A}{a_2 = a_1 : A} \qquad \frac{a_1 = a_2 : A \qquad a_2 = a_3 : A}{a_1 = a_3 : A}$$

$$\frac{a : A_1 \qquad A_1 = A_2}{a : A_2} \qquad \frac{a_1 = a_2 : A_1 \qquad A_1 = A_2}{a_1 = a_2 : A_2}$$

Substitution rule.

$$\frac{(x : A, \Delta)\ \mathcal{B} \qquad a : A}{(\Delta[a/x])\ \mathcal{B}[a/x]}$$

Congruence rules.

$$\frac{(x : A, \Delta)\ C : \mathrm{type} \qquad a_1 = a_2 : A}{(\Delta[a_1/x])\ C[a_1/x] = C[a_2/x] : \mathrm{type}} \qquad \frac{(x : A, \Delta)\ c : C \qquad a_1 = a_2 : A}{(\Delta[a_1/x])\ c[a_1/x] = c[a_2/x] : C[a_1/x]}$$

For ∇ that is either Π, Σ or W:

$$\frac{(x : A)\ B_1 = B_2 : \mathrm{type}}{(\nabla x : A)B_1 = (\nabla x : A)B_2 : \mathrm{type}} \qquad \frac{(x : A)\ b_1 = b_2 : B}{(\lambda x : A)b_1 = (\lambda x : A)b_2 : (\Pi x : A)B}$$

Analogous rules should also be formulated for other symbols, but we omit for brevity.

Basic types. We write $\mathbb{0}$, $\mathbb{1}$, and $\mathbb{2}$ for the types with zero, one, and two canonical elements, respectively. We thus have the following formation and introduction rules.

$$\mathbb{0} : \mathrm{type} \qquad \mathbb{1} : \mathrm{type} \qquad \mathbb{2} : \mathrm{type}$$

The type with no canonical elements does not have an introduction rule. We thus have only the following introduction rules.

$$0_1 : \mathbb{1} \qquad 1_2 : \mathbb{2} \qquad 2_2 : \mathbb{2}$$

The elimination rules follow the usual pattern.

$$\frac{(z : \mathbb{0})\ C : \mathrm{type} \qquad e : \mathbb{0}}{\mathrm{r}_0(e) : C[e/z]}$$

$$\frac{(z : \mathbb{1})\ C : \mathrm{type} \qquad e : \mathbb{1} \qquad c : C[0_1/z]}{\mathrm{r}_1(e, c) : C[e/z]}$$

$$\frac{(z : \mathbb{2})\ C : \mathrm{type} \qquad e : \mathbb{2} \qquad c_1 : C[1_2/z] \qquad c_2 : C[2_2/z]}{\mathrm{r}_2(e, c_1, c_2) : C[e/z]}$$

Finally, we give the computation rules for these types.

$$\frac{(z : \mathbb{1})\ C : \mathrm{type} \qquad c : C[0_1/z]}{\mathrm{r}_1(0, c) = c : C[0_1/z]}$$

$$\frac{(z : \mathbb{2})\ C : \mathrm{type} \qquad c_1 : C[1_2/z] \qquad c_2 : C[2_2/z]}{\mathrm{r}_2(1_2, c_1, c_2) = c_1 : C[1_2/z]}$$

$$\frac{(z : \mathbb{2})\ C : \mathrm{type} \qquad c_1 : C[1_2/z] \qquad c_2 : C[2_2/z]}{\mathrm{r}_2(2_2, c_1, c_2) = c_2 : C[2_2/z]}$$

Natural numbers type.

$$\mathbb{N} : \mathrm{type}$$

The introduction rules for this type are standard.

$$0_{\mathbb{N}} : \mathbb{N} \qquad \frac{n : \mathbb{N}}{\mathrm{succ}(n) : \mathbb{N}}$$

In the elination and computation rules below the premisses should include the judgement $(z : \mathbb{N})\ C : \mathrm{type}$ that we omit for brevity.

$$\frac{n : \mathbb{N} \quad c : C[0/z] \quad (x : \mathbb{N}, y : C[x/z])\ d : C[\mathrm{succ}(x)/z]}{\mathrm{r}_{\mathbb{N}}(n, c, (x,y)d) : C[n/z]}$$

$$\frac{0 : \mathbb{N} \quad c : C[0/z] \quad (x : \mathbb{N}, y : C[x/z])\ d : C[\mathrm{succ}(x)/z]}{\mathrm{r}_{\mathbb{N}}(0, c, (x,y)d) = c : C[0/z]}$$

$$\frac{n : \mathbb{N} \quad c : C[0/z] \quad (x : \mathbb{N}, y : C[x/z])\ d : C[\mathrm{succ}(x)/z]}{\mathrm{r}_{\mathbb{N}}(\mathrm{succ}(n), c, (x,y)d) = d[n, \mathrm{r}_{\mathbb{N}}(n, c, (x,y)d)/x, y] : C[\mathrm{succ}(n)/z]}$$

$\mathbb{R}_2$*-rules.*

$$\frac{e : 2 \qquad A_1 : \mathrm{type} \qquad A_2 : \mathrm{type}}{\mathbb{R}_2(e, A_1, A_2) : \mathrm{type}}$$

$$\mathbb{R}_2(1_2, A_1, A_2) = A_1 \qquad \mathbb{R}_2(2_2, A_1, A_2) = A_2$$

Σ*-rules.*

$$\frac{A : \mathrm{type} \qquad (x : A)\ B : \mathrm{type}}{(\Sigma x : A)B : \mathrm{type}}$$

$$\frac{a : A \qquad b : B[a/x]}{\mathrm{pair}(a,b) : (\Sigma x : A)B}$$

Similarly to what we did for the natural numbers type, we suppress the premiss $(z : (\Sigma x : A))\ C : \mathrm{type}$ in the elimination and computation rules.

$$\frac{e : (\Sigma x : A)B \quad (x : A, y : B)\ c : C[\mathrm{pair}(x,y)/z]}{\mathrm{split}(e, (x,y)c) : C[e/z]}$$

$$\frac{a : A \quad b : B[a/x] \quad (x : A, y : B)\ c : C[\mathrm{pair}(x,y)/z]}{\mathrm{split}((\mathrm{pair}(a,b), (x,y)c) = c[a, b/x, y] : C[\mathrm{pair}(a,b)/z]}$$

Π-*rules.*

$$\frac{A : \mathrm{type} \qquad (x : A)\ B : \mathrm{type}}{(\Pi x : A)B : \mathrm{type}}$$

$$\frac{(x : A)\ b : B}{(\lambda x : A)b : (\Pi x : A)B}$$

$$\frac{f : (\Pi x : A)B \qquad a : A}{\mathrm{app}(f, a) : B[a/x]}$$

$$\frac{(x : A)\ b : B \qquad a : A}{\mathrm{app}((\lambda x : A)b, a) = b[a/x] : B[a/x]}$$

Rules for the type universe.

Formation rule.

$$\mathbb{U} : \mathrm{type}$$

Introduction rules.

$$\widehat{\mathbb{0}} : \mathbb{U} \qquad \widehat{\mathbb{1}} : \mathbb{U} \qquad \widehat{\mathbb{2}} : \mathbb{U} \qquad \widehat{\mathbb{N}} : \mathbb{U}$$

$$\frac{e : \mathbb{2} \quad a_1 : \mathbb{U} \quad a_2 : \mathbb{U}}{\mathbb{R}_2(e, a_1, a_2) : \mathbb{U}}$$

$$\frac{a : \mathbb{U} \qquad (x : \mathbb{T}(a))\ b : \mathbb{U}}{(\Sigma x : a)b : \mathbb{U}} \qquad \frac{a : \mathbb{U} \qquad (x : \mathbb{T}(a))\ b : \mathbb{U}}{(\Pi x : a)b : \mathbb{U}}$$

Elimination rule.

$$\frac{a : \mathbb{U}}{\mathbb{T}(a) : \mathrm{type}}$$

Computation rules.

$$\mathbb{T}(\widehat{\mathbb{0}}) = \mathbb{0} : \mathrm{type} \quad \mathbb{T}(\widehat{\mathbb{1}}) = \mathbb{1} : \mathrm{type} \quad \mathbb{T}(\widehat{\mathbb{2}}) = \mathbb{2} : \mathrm{type} \quad \mathbb{T}(\widehat{\mathbb{N}}) = \mathbb{N} : \mathrm{type}$$

$$\frac{e : \mathbb{2} \qquad a_1 : \mathbb{U} \qquad a_2 : \mathbb{U}}{\mathbb{T}(\mathbb{R}_2(e, a_1, a_2)) = \mathbb{R}_2(e, \mathbb{T}(a_1), \mathbb{T}(a_2)) : \mathrm{type}}$$

$$\frac{a : \mathbb{U} \qquad (x : \mathbb{T}(a))\ b : \mathbb{U}}{\mathbb{T}((\Sigma x : a)b) = (\Sigma x : \mathbb{T}(a))\,\mathbb{T}\,b : \mathrm{type}}$$

$$\frac{a : \mathbb{U} \qquad (x : \mathbb{T}(a))\ b : \mathbb{U}}{\mathbb{T}((\Pi x : a)b) = (\Pi x : \mathbb{T}(a))\,\mathbb{T}(b) : \mathrm{type}}$$

Rules for the proposition universe.

Formation rule.

$$\mathbb{P} : \mathrm{type}$$

Introduction rules.

$$\widehat{\top} : \mathbb{P} \qquad \widehat{\bot} : \mathbb{P}$$

$$\frac{p_1 : \mathbb{P} \qquad p_2 : \mathbb{P}}{p_1 \wedge p_2 : \mathbb{P}} \qquad \frac{p_1 : \mathbb{P} \qquad p_2 : \mathbb{P}}{p_1 \vee p_2 : \mathbb{P}} \qquad \frac{p_1 : \mathbb{P} \qquad p_2 : \mathbb{P}}{p_1 \supset p_2 : \mathbb{P}}$$

$$\frac{a : \mathbb{U} \qquad (x : \mathbb{T}(a))\ p : \mathbb{P}}{(\forall x : a)p : \mathbb{P}} \qquad \frac{a : \mathbb{U} \qquad (x : \mathbb{T}(a))\ p : \mathbb{P}}{(\exists x : a)p : \mathbb{P}}$$

Elimination rule.

$$\frac{p : \mathbb{P}}{\tau(p) : \mathrm{prop}}$$

Computation rules.

$$\tau(\widehat{\top}) \equiv \top \qquad \tau(\widehat{\bot}) \equiv \bot$$

$$\frac{p_1 : \mathbb{P} \qquad p_2 : \mathbb{P}}{\tau(p_1 \wedge p_2) \equiv \tau(p_1) \wedge \tau(p_2)} \qquad \frac{p_1 : \mathbb{P} \qquad p_2 : \mathbb{P}}{\tau(p_1 \vee p_2) \equiv \tau(p_1) \vee \tau(p_2)}$$

$$\frac{p_1 : \mathbb{P} \qquad p_2 : \mathbb{P}}{\tau(p_1 \supset p_2) \equiv \tau(p_1) \supset \tau(p_2)}$$

$$\frac{a : \mathbb{U} \qquad (x : \mathbb{T}(a))\ p : \mathbb{P}}{\tau((\forall x : a)p) \equiv (\forall x : \mathbb{T}(a))\,\tau(p)} \qquad \frac{a : \mathbb{U} \qquad (x : \mathbb{T}(a))\ p : \mathbb{P}}{\tau((\exists x : a)p) \equiv (\exists x : \mathbb{T}(a))\,\tau(p)}$$

References

[1] P. Aczel. The type theoretic interpretation of constructive set theory. In A. MacIntyre, L. Pacholski, and J. Paris, editors, *Logic Colloquium '77*, pages 55 – 66. North-Holland, 1978.

[2] P. Aczel. The type theoretic interpretation of constructive set theory: choice principles. In A. S. Troelstra and D. van Dalen, editors, *The L. E. J. Brouwer Centenary Symposium*, pages 1 – 40. North-Holland, 1982.

[3] P. Aczel. The type theoretic interpretation of constructive set theory: inductive definitions. In R. Barcan Marcus, G.J.W. Dorn, and P. Weinegartner, editors, *Logic, Methodology and Philosophy of Science VII*, pages 17 – 49. North-Holland, 1986.

[4] P. Aczel and N. Gambino. Collection principles in Dependent Type Theory. In P. Callaghan, Z. Luo, J. McKinna, and R. Pollack, editors, *Types for Proofs and Programs*, volume 2277 of *Lecture Notes in Computer Science*, pages 1 – 23. Springer, 2002.

[5] P. Aczel and M. Rathjen. Notes on Constructive Set Theory. Technical Report 40, Mittag-Leffler Institute, The Swedish Royal Academy of Sciences, 2001. Available from the first author's web page at `http://guat.a.com/cm..pwww.cs.man.ac.uk/~petera/papers.html`.

[6] S. Awodey and A. Bauer. Propositions as [types]. *Journal of Logic and Computation*, 14(4):447 – 471, 2004.
[7] T. Coquand, G. Sambin, J. M. Smith, and S. Valentini. Inductively generated formal topologies. *Annals of Pure and Applied Logic*, 124(1-3):71–106, 2003.
[8] H. M. Friedman. The consistency of classical set theory relative to a set theory with intuitionistic logic. *Journal of Symbolic Logic*, 38:315–319, 1973.
[9] H. M. Friedman. Set theoretic foundations of constructive analysis. *Annals of Mathematics*, 105:1–28, 1977.
[10] N. Gambino. Types and sets — A study on the jump to full impredicativity, 1999. Laurea dissertation, Department of Pure and Applied Mathematics, University of Padua.
[11] N. Gambino. *Sheaf interpretations for generalised predicative intuitionistic systems*. PhD thesis, University of Manchester, 2002.
[12] N. Gambino. Heyting-valued interpretations for Constructive Set Theory. *Annals of Pure and Applied Logic*, To appear.
[13] N. Gambino. Presheaf models for constructive set theory. In L. Crosilla and P. Schuster, editors, *From sets and types to topology and analysis*. Oxford University Press, To appear.
[14] R. J. Grayson. Heyting-valued models for Intuitionistic Set Theory. In M. P. Fourman, C. J. Mulvey, and D. S. Scott, editors, *Applications of Sheaves*, volume 753 of *Lecture Notes in Mathematics*, pages 402 – 414. Springer, 1979.
[15] E. R. Griffor and M. Rathjen. The strength of some Martin-Löf type theories. *Archive for Mathematical Logic*, 33:347 – 385, 1994.
[16] B. Jacobs. *Categorical Logic and Type Theory*. North-Holland, 1999.
[17] P. T. Johnstone. *Stone Spaces*. Cambridge University Press, 1982.
[18] R. S. Lubarsky. Independence results around constructive ZF. *Annals of Pure and Applied Logic*, 132(2-3):209 – 225, 2005.
[19] S. MacLane and I. Moerdijk. *Sheaves in Geometry and Logic — A First Introduction to Topos Theory*. Springer, 1992.
[20] M. E. Maietti and G. Sambin. Towards a minimalistic foundation for constructive mathematics. In L. Crosilla and P. Schuster, editors, *From Sets and Types to Topology and Analysis*. Oxford University Press, To appear.
[21] P. Martin-Löf. *Intuitionistic Type Theory — Notes by G. Sambin of a series of lectures given in Padua, June 1980*. Bibliopolis, 1984.
[22] I. Moerdijk and E. Palmgren. Wellfounded trees in categories. *Journal of Pure and Applied Logic*, 104:189 – 218, 2000.
[23] I. Moerdijk and E. Palmgren. Type theories, toposes and Constructive Set Theory: predicative aspects of AST. *Annals of Pure and Applied Logic*, 114(1-3):155–201, 2002.
[24] J.R. Myhill. Constructive Set Theory. *Journal of Symbolic Logic*, 40(3):347–382, 1975.
[25] B. Nordström, K. Petersson, and J. M. Smith. Martin-Löf Type Theory. In S. Abramski, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 5. Oxford University Press, 2000.
[26] M. Rathjen. CZF has the disjunction and the numerical existence property. Available from the author's web page, 2004.
[27] M. Rathjen. Realizability for Constructive Zermelo-Fraenkel Set Theory. Available from the author's web page, 2004.
[28] M. Rathjen. Replacement versus Collection in Constructive Zermelo-Fraenkel Set Theory. Available from the author's web page, 2004.
[29] M. Rathjen and R. S. Lubarsky. On the regular extension axiom and its variants. *Mathematical Logic Quarterly*, 49(5):511–518, 2003.

LaCIM et CIRGET, Université du Québec à Montréal, Case Postale 8888, Succursale Centre-Ville, Montréal (Québec) H3C 3P8, Canada
E-mail address: gambino@math.uqam.ca

Schools of Mathematics and Computer Science, University of Manchester, Manchester M13 9PL, England
E-mail address: petera@cs.man.ac.uk