

INTERPRETATION OF INACCESSIBLE SETS IN MARTIN-LÖF TYPE THEORY WITH ONE MAHLO UNIVERSE

YUTA TAKAHASHI 

Aomori University, 2-3-1 Kobata, Aomori City, Aomori 030-0943, Japan
e-mail address: y.takahashi@aomori-u.ac.jp

ABSTRACT. Rathjen proved that Aczel’s constructive set theory **CZF** extended with inaccessible sets of all transfinite orders can be interpreted in Martin-Löf type theory **MLTT** extended with Setzer’s Mahlo universe and another universe above it. In this paper we show that this interpretation can be carried out bottom-up without the universe above the Mahlo universe, provided we add an accessibility predicate instead. If we work in Martin-Löf type theory with extensional identity types the accessibility predicate can be defined in terms of W-types. The main part of our interpretation has been formalised in the proof assistant Agda.

1. INTRODUCTION

Background and Aim. Martin-Löf type theory **MLTT** was extended by Setzer [Set00, Set08b] with the so-called Mahlo universe types to provide a variant of intensional **MLTT** equipped with an analogue of Mahlo cardinals. Specifically, a Mahlo universe $\mathbb{M}$ has a reflection property similar to that of Mahlo cardinals [Mah11] (see also, *e.g.*, [Dra74]): for any function f of type $\Sigma_{(x:\mathbb{M})}(\mathrm{T}_{\mathbb{M}}x \rightarrow \mathbb{M}) \rightarrow \Sigma_{(x:\mathbb{M})}(\mathrm{T}_{\mathbb{M}}x \rightarrow \mathbb{M})$, where $\mathrm{T}_{\mathbb{M}}$ is the decoding function for $\mathbb{M}$, the universe $\mathbb{M}$ contains the subuniverse which is closed under f and denoted by U_f . The resulting system, **MLM**, is thus an instance of a constructive system extended with an analogue of Mahlo cardinals. Setzer’s purpose of introducing **MLM** is to obtain an extension of **MLTT** whose proof-theoretic ordinal is slightly greater than that of **KPM**, *i.e.*, Kripke-Platek set theory with one recursively Mahlo ordinal, where the proof-theoretic ordinals of a system enable measurement of the proof-theoretic strength. The proof-theoretic ordinal of **MLM** was determined by the results of [Set00, Set08b].

Another instance of constructive systems extended with an analogue of Mahlo cardinals was developed in the context of Aczel’s constructive set theory: **CZF** [Acz78, Acz82, Acz86]. Rathjen [Rat03] formulated the system **CZFM** by extending **CZF** with the axiom asserting the existence of one Mahlo set. He then verified that **CZFM** is interpretable in an extension of **MLM** which has one usual Tarski universe above the Mahlo universe $\mathbb{M}$. Though he calls this extension **MLM**, Setzer’s **MLM** does not have any universe above $\mathbb{M}$ and so we call this extension **MLM**⁺ to distinguish it from **MLM**. The main purpose of introducing **CZFM**

Key words and phrases: Martin-Löf type theory, constructive set theory, Mahlo universe types.

is a proof-theoretic one, as in the case of Mahlo universes. The interpretation of **CZFM** in $\mathbf{MLM}^+$ implies that **CZFM** has at most the proof-theoretic strength of $\mathbf{MLM}^+$.

On the other hand, Rathjen, Griffor and Palmgren [RGP98] introduced the system $\mathbf{CZF}_\pi$, which is an extension of **CZF** with the inaccessible sets of all transfinite orders [Mah11]. They also introduced an extension of **MLTT** with second-order universe operators called **MLQ**, where the base theory of **MLQ** is extensional **MLTT** in [ML84]. Then, Rathjen, Griffor and Palmgren showed that $\mathbf{CZF}_\pi$ has at most the proof-theoretic strength of **MLQ** by verifying that $\mathbf{CZF}_\pi$ is interpretable in **MLQ**. As to the strength of large sets concerned, $\mathbf{CZF}_\pi$ is weaker than **CZFM**: this claim is verified by Rathjen in [Rat03], where he proved that $\mathbf{CZF}_\pi$ is interpretable in **CZFM**.

In summary, Rathjen [Rat03] have shown that $\mathbf{CZF}_\pi$ can be interpreted in $\mathbf{MLM}^+$ by proving that $\mathbf{CZF}_\pi$ and **CZFM** are interpretable in **CZFM** and $\mathbf{MLM}^+$, respectively. However, it was unknown whether a universe above the Mahlo universe $\mathbb{M}$ is indispensable for the interpretation of $\mathbf{CZF}_\pi$ in **MLTT** with the Mahlo universe $\mathbb{M}$. Rathjen used the universe above $\mathbb{M}$ to define the type of Aczel’s iterative sets, which is a model of **CZFM** in $\mathbf{MLM}^+$. So the universe above $\mathbb{M}$ is crucial to his argument, and adding one universe type makes a system stronger as shown by Setzer [Set98, Set93]. Are the inaccessible sets of $\mathbf{CZF}_\pi$ interpretable by $\mathbb{M}$ without a universe above it?

In this paper, we show that $\mathbf{CZF}_\pi$ is interpretable in **MLM** with the accessibility predicate *Acc*: one can dispense with a universe above the Mahlo universe, but the trade-off is the use of *Acc*. The accessibility predicate *Acc* provides the transfinite induction principle on Aczel’s iterative sets. This predicate can be defined in Dybjer-Setzer’s theory of induction-recursion [DS06], which is a finite axiomatisation of Dybjer’s general schema of simultaneous inductive-recursive definitions [Dyb00]. Specifically, one can define *Acc* by indexed inductive definition. To the best of our knowledge, **MLTT** without the schema of indexed inductive definition is not sufficient to define indexed inductive types such as *Acc*, though some progress was made by, *e.g.*, Hugunin [Hug21]. We thus add *Acc* as a base type to **MLM**, while we show in Appendix B that *extensional MLTT* in [ML84] can define *Acc*. So, in the extensional setting, *Acc* is not necessary in our interpretation of $\mathbf{CZF}_\pi$ as in that of [RGP98].

Contrary to Rathjen’s elegant but “top-down” argument, the Mahlo universe $\mathbb{M}$ together with *Acc* enables to provide a “bottom-up” argument to interpret $\mathbf{CZF}_\pi$. Below we first define a higher-order (in fact, second-order) universe operator using the reflection property of $\mathbb{M}$ in **MLM**. Next, the transfinite hierarchy of Aczel’s iterative sets is constructed by iterating this universe operator along the accessibility given by *Acc*. Finally, we verify that these sets are indeed type-theoretic counterparts of inaccessible sets. Besides this aim, we provide an explicit construction of super universes as subuniverses of $\mathbb{M}$ for the expository purpose. In particular, it will be shown that the version of super-universe construction which is given by the higher-order universe operator above can be iterated.

Moreover, we formalised the main part of our proof in the proof assistant Agda [Agd25]. The Mahlo universe $\mathbb{M}$ is simulated by the *external* Mahlo universe in Agda: the external Mahlo universe is a proof-theoretically weak variant of $\mathbb{M}$, and can be defined with induction-recursion as Dybjer and Setzer have shown [DS03]. So Agda is equipped with the external Mahlo universe because Agda supports even indexed induction-recursion, which is a generalisation of induction-recursion. This means that Agda also has the accessibility predicate *Acc*. In Agda we defined the higher-order universe operator above by using the reflection property of the external Mahlo universe, and formulated our type-theoretic

Acronym	Description
MLM	intensional Martin-Löf type theory with Setzer's Mahlo universe [Set00, Set08b]
MLM ⁺	MLM with a universe above its Mahlo universe [Rat03]
MLQ	extensional Martin-Löf type theory with second-order universe operators [RGP98]
MLM _{acc}	MLM with an accessibility predicate
MLM _{ext}	MLM with extensional identity types
CZF	Aczel's constructive Zermelo-Fraenkel set theory [Acz78, Acz82, Acz86]
CZF _π	CZF with inaccessible sets of all transfinite orders [RGP98]
CZFM	CZF with one Mahlo set [Rat03]

Table 1: The Systems Discussed in the Present Paper

interpretation of **CZF**_π. At the end of Section 3.2, we will comment on the part of our proof which is not formalised yet.

Note that neither our argument in this paper nor our formalisation postulates extensional principles such as the rules of extensional identity types, except that Appendix B derives all inference rules for Acc by using extensional **MLTT** in [ML84]. The Agda code for the formalisation is available online [Tak25]. The clickable symbol [Link](#), which is attached to some statements in the main part of our proof, will open the Agda code in [Tak25] corresponding to those statements. All files in this repository can be typechecked with Agda version 2.7.0.1.

In Table 1 we list all of the systems discussed in this paper.

Related Work. The interpretation of **CZF** in **MLTT** was first discussed in [Acz78]. Here Aczel set out **CZF** and interpreted it in the fragment **ML**₁**V** of **MLTT**, which has dependent function types (*i.e.*, Π -types), dependent pair types (*i.e.*, Σ -types), disjoint union types, the empty type, the unit type, the natural number type, the universe type and an instance of W-types. Iterative sets in **CZF** were interpreted by this instance of W-types. Later, the set-theoretic axiom of dependent choice was interpreted in the extension of **ML**₁**V** by extensional identity types [Acz82]. Moreover, the regular extension axiom (**REA**), which states that any set is a subset of a regular set, was interpreted by further adding W-types [Acz86].

Griffor and Rathjen [GR94] showed that **CZF** and **ML**₁**V** have the same proof-theoretic strength and there is a fragment of **MLTT** with the same proof-theoretic strength as **CZF** + **REA**. Setzer [Set98, Set93] determined the proof-theoretic strength of the fragment of **MLTT** obtained from **ML**₁**V** by adding intensional identity types and W-types: he computed its proof-theoretic ordinal without appealing to Aczel's interpretation of **CZF** in **MLTT**.

As aforementioned, Rathjen, Griffor and Palmgren [RGP98] introduced the extension **CZF**_π of **CZF** by the axiom asserting the existence of inaccessible sets of all transfinite orders. The type system in which **CZF**_π was interpreted is **MLQ**, which is obtained from extensional **MLTT** in [ML84] by adding the inductive type Q and universe type M: Q is an inductive type of codes for operators which provides universes closed under universe operators constructed previously, and M is a universe closed under the operators whose codes belong to Q. Our higher-order universe operator is inspired by the one defined with the types Q and M, and our interpretation of **CZF**_π is the same as that of [RGP98], except

for the interpretation of inaccessible sets. While **MLM** is proof-theoretically stronger than **MLQ**, it is an open question whether the intensional variant of **MLQ** can be interpreted in **MLM**. There is a difficulty in simulating the type of *codes* for operators of **MLQ** by **MLM** because such a type of codes for higher-order objects is absent from **MLM**. Rathjen [Rat00] introduced the fragment **MLF** of **MLQ** and showed that two weaker variants of **CZF**_π have the same proof-theoretic strength as **MLF**.

The relationship between **MLTT** and **CZF** in the context of an analogue for Mahlo cardinals has already been examined by Rathjen [Rat03], as shown above. He introduced the notion of a Mahlo set into **CZF** and verified that the resulting system **CZFM** is interpretable in **MLM**⁺, which is an extension of **MLM** with one Tarski universe above the Mahlo universe. He also showed that **CZF**_π is interpretable in **CZFM**, but this result is not explicitly stated and so we briefly recall it in Section 3.1.

Contributions. Our contributions are summarised as follows:

- (1) The principal contribution of this paper is the construction of inaccessible sets of transfinite orders in **MLM** with the accessibility predicate *Acc*. To describe this interpretation, we use both the reflection property of **M** and the accessibility predicate *Acc*. We define a higher-order universe operator with the reflection property of **M** and iterate this operator along the accessibility given by *Acc*.
- (2) A secondary contribution is the Agda formalisation of the main proof for the result above. We utilise indexed induction-recursion in Agda for formalisation: the Mahlo universe of **MLM** is simulated by the external Mahlo universe defined with induction-recursion, and the predicate *Acc* is defined by using indexed inductive definition.

Organisation of the Paper. In Section 2, we describe the system **MLM** and define a higher-order universe operator using the reflection property of the Mahlo universe **M**. Section 3 first provides a brief introduction to **CZF**_π and then shows that **CZF**_π is interpretable in **MLM** with the accessibility predicate *Acc*. Specifically, we iterate the application of our higher-order operator along the accessibility given by *Acc* to construct the hierarchy of inaccessible sets in **MLM**.

2. A HIGHER-ORDER UNIVERSE OPERATOR BY THE MAHLO UNIVERSE IN **MLM**

In this section, we first present the system **MLM** of Martin-Löf type theory with one Mahlo universe (Section 2.1). Next, we formulate a higher-order universe operator using the reflection property of the Mahlo universe (Section 2.2). This operator will be used to construct the transfinite hierarchy of inaccessible sets.

2.1. Martin-Löf Type Theory with One Mahlo Universe. We follow Setzer’s work [Set00, Set08b, Set08a] in defining the rules of universes in the system **MLM** of Martin-Löf type theory with one Mahlo universe, while we follow [Hof97, Set98] in defining the inference rules other than those of universes. The system **MLM** is an intensional **MLTT** (*i.e.*, an **MLTT** with the intensional identity types) formulated in the polymorphic way. See [NPS90, DP20] for the distinction between intensional **MLTT**’s and extensional ones; see [NPS90] for the distinction between the polymorphic formulation and the monomorphic one.

$$\begin{aligned}
a, b, c, d, f, g, h, l ::= & x \mid 0 \mid m_n \mid \lambda x. a \mid \mathbf{app}(a, b) \mid (a, b) \mid E_\Sigma(a, b) \mid \\
& i(a) \mid j(a) \mid E_+(a, b, c) \mid \mathbf{suc}(a) \mid E_N(a, b, c) \mid \\
& E_n(a_0, \dots, a_n) \mid \mathbf{sup}(a, b) \mid E_W(a, b) \mid \mathbf{refl}(a) \mid E_{\text{Id}}(a, b) \mid \\
& \widehat{N}_{\mathbb{M}} \mid \widehat{N}_{n\mathbb{M}} \mid \widehat{\Pi}_{\mathbb{M}}(a, b) \mid \widehat{\Sigma}_{\mathbb{M}}(a, b) \mid a \widehat{+}_{\mathbb{M}} b \mid \widehat{W}_{\mathbb{M}}(a, b) \mid \\
& \widehat{\text{Id}}_{\mathbb{M}}(a, b, c) \mid \widehat{U}_f \mid \widehat{T}_f(a) \mid \mathbf{res}_f^0(a, b) \mid \mathbf{res}_f^1(a, b, c) \\
& \widehat{N}_f \mid \widehat{N}_{nf} \mid \widehat{\Pi}_f(a, b) \mid \widehat{\Sigma}_f(a, b) \mid a \widehat{+}_f b \mid \widehat{W}_f(a, b) \mid \widehat{\text{Id}}_f(a, b, c) \\
A, B, C, D ::= & N \mid N_n \mid \mathbb{M} \mid U_f \mid \Pi_{(x:A)} B \mid \Sigma_{(x:A)} B \mid A + B \mid \\
& W_{(x:A)} B \mid \text{Id}(A, a, b) \mid T_{\mathbb{M}}(a)
\end{aligned}$$

Table 2: Pre-Terms and Pre-Types

In this paper, syntactic variables are used possibly with suffixes. Let an infinite set of variables be given. Variables are denoted by x, y, z, v, w , and numerals $0, 1, \dots$ are denoted by n, m, k . *Pre-terms* $a, b, c, \dots$ and *pre-types* $A, B, C, \dots$ are defined by simultaneous induction in Table 2. The intuitive meanings of pre-terms and pre-types will be explained by the inference rules of **MLM**. The pre-terms of the form $\mathbf{app}(a, b)$ are abbreviated as $a b$. When a variable x is not free in B , we write $\Pi_{(x:A)} B$ as $A \rightarrow B$, and $\Sigma_{(x:A)} B$ as $A \times B$. The occurrences of x are bound in the expressions of the form $\Pi_{(x:A)} B, \Sigma_{(x:A)} B, W_{(x:A)} B$ or $\lambda x. a$. Consecutive applications $k_1(k_2(\dots k_n(a) \dots))$ of i, j with $k_1, k_2, \dots, k_n \in \{i, j\}$ are abbreviated as $k_1 k_2 \dots k_n a$. Any two α -equivalent expressions are identified. In addition, we adopt Barendregt's variable condition: no variable occurs both as a free one and as a bound one in an expression, and all bound variables in an expression are distinct. We denote by $e[e_1/x_1, \dots, e_n/x_n]$ the expression obtained by substituting $e_1, \dots, e_n$ for $x_1, \dots, x_n$ in e consecutively. We always assume that no bound variable in e occurs freely in any of $e_1, \dots, e_n$, by applying the α -conversion if necessary.

Judgements are expressions which have one of the following forms:

- $\vdash \Gamma$ ctxt, which means that Γ is a valid context,
- $\Gamma \vdash A$ type, which means that A is a type in a context Γ ,
- $\Gamma \vdash a : A$, which means that a is a term of type A in a context Γ ,
- $\Gamma \vdash A = B$, which means that A and B are definitionally equal types in a context Γ ,
- $\Gamma \vdash a = b : A$, which means that a and b are definitionally equal terms of type A in a context Γ .

According to the propositions-as-types conception, $\Gamma \vdash a : A$ means that a is a proof of the proposition A in the context Γ . One can consider two definitionally equal expressions to be two expressions whose results of computation are equal. In an extensional **MLTT** a judgement of the form $\Gamma \vdash A = B$ or $\Gamma \vdash a = b : A$ no longer means that the two expressions mentioned are definitionally equal: an extensional **MLTT** treats these judgement forms in a more extensional manner (for the rules of extensional identity types, see Appendix B). We denote by θ an expression of one of the following forms:

$$A \text{ type}, \quad a : A, \quad A = B, \quad a = b : A.$$

The system **MLM** is a formal system to derive judgements of the forms above by means of inference rules. Its inference rules are classified as follows.

- General rules: those of contexts, assumptions, substitutions and the definitional equality,
- Rules for standard types: those of Π -types, Σ -types, $+$ -types, N -type, N_n -types, W -types and Id -types,
- Rules for universe types: those of Mahlo universe types and non-Mahlo universe types.

Each collection of rules for standard types includes the *formation rule*, the *introduction rules*, the *elimination rules* and the *equality rules*. The formation rule explains how to form the types, and the introduction rules enable the construction of canonical elements of the types. On the other hand, the elimination rules provide executable elements of the types, and the equality rules specify how to execute (or compute) these elements in an equational way. Here we present the rules for Mahlo universe only, and the other rules of **MLM** are presented in Appendix A. For detailed explanations of the above rules except those of Mahlo universes, see, *e.g.*, [ML84, NPS90, Hof97, DP20]. Below we denote the Mahlo universe in **MLM** by $\mathbb{M}$, while it is denoted by V in [Set00, Set08b, Set08a].

For the Mahlo Universe $(\mathbb{M}, T_{\mathbb{M}})$, we have the $\mathbb{M}$ -formation rule $\mathbb{M}\text{fm}$, the $\mathbb{M}$ -introduction rules $\mathbb{M}\text{in}$ and the $T_{\mathbb{M}}$ -formation rule $T_{\mathbb{M}}\text{fm}$ with the inference rules which define the decoding function $T_{\mathbb{M}}$ recursively. Congruence rules are defined for the $T_{\mathbb{M}}$ -formation rule and all $\mathbb{M}$ -introduction rules except $\mathbb{M}\text{in}_N$ and $\mathbb{M}\text{in}_{N_n}$, though we omit to write down them. Some examples of congruence rules can be found in the list of the rules for Π -types in Appendix A.

$$\begin{array}{c}
\frac{\vdash \Gamma \text{ ctxt}}{\Gamma \vdash \mathbb{M} \text{ type}} \mathbb{M}\text{fm} \qquad \frac{\Gamma \vdash a : \mathbb{M}}{\Gamma \vdash T_{\mathbb{M}}(a) \text{ type}} T_{\mathbb{M}}\text{fm} \\[10pt]
\frac{\vdash \Gamma \text{ ctxt}}{\Gamma \vdash \widehat{N}_{\mathbb{M}} : \mathbb{M}} \mathbb{M}\text{in}_N \qquad \frac{\vdash \Gamma \text{ ctxt}}{\Gamma \vdash T_{\mathbb{M}}(\widehat{N}_{\mathbb{M}}) = N} \\[10pt]
\frac{\vdash \Gamma \text{ ctxt}}{\Gamma \vdash \widehat{N}_{n\mathbb{M}} : \mathbb{M}} \mathbb{M}\text{in}_{N_n} \qquad \frac{\vdash \Gamma \text{ ctxt}}{\Gamma \vdash T_{\mathbb{M}}(\widehat{N}_{n\mathbb{M}}) = N_n} \\[10pt]
\frac{\Gamma \vdash a : \mathbb{M} \quad \Gamma \vdash b : T_{\mathbb{M}}(a) \rightarrow \mathbb{M}}{\Gamma \vdash \widehat{\Pi}_{\mathbb{M}}(a, b) : \mathbb{M}} \mathbb{M}\text{in}_{\Pi} \qquad \frac{\Gamma \vdash a : \mathbb{M} \quad \Gamma \vdash b : T_{\mathbb{M}}(a) \rightarrow \mathbb{M}}{\Gamma \vdash T_{\mathbb{M}}(\widehat{\Pi}_{\mathbb{M}}(a, b)) = \Pi_{(x:T_{\mathbb{M}}(a))} T_{\mathbb{M}}(b x)} \\[10pt]
\frac{\Gamma \vdash a : \mathbb{M} \quad \Gamma \vdash b : T_{\mathbb{M}}(a) \rightarrow \mathbb{M}}{\Gamma \vdash \widehat{\Sigma}_{\mathbb{M}}(a, b) : \mathbb{M}} \mathbb{M}\text{in}_{\Sigma} \qquad \frac{\Gamma \vdash a : \mathbb{M} \quad \Gamma \vdash b : T_{\mathbb{M}}(a) \rightarrow \mathbb{M}}{\Gamma \vdash T_{\mathbb{M}}(\widehat{\Sigma}_{\mathbb{M}}(a, b)) = \Sigma_{(x:T_{\mathbb{M}}(a))} T_{\mathbb{M}}(b x)} \\[10pt]
\frac{\Gamma \vdash a : \mathbb{M} \quad \Gamma \vdash b : \mathbb{M}}{\Gamma \vdash a \widehat{+}_{\mathbb{M}} b : \mathbb{M}} \mathbb{M}\text{in}_{+} \qquad \frac{\Gamma \vdash a : \mathbb{M} \quad \Gamma \vdash b : \mathbb{M}}{\Gamma \vdash T_{\mathbb{M}}(a \widehat{+}_{\mathbb{M}} b) = T_{\mathbb{M}}(a) + T_{\mathbb{M}}(b)} \\[10pt]
\frac{\Gamma \vdash a : \mathbb{M} \quad \Gamma \vdash b : T_{\mathbb{M}}(a) \rightarrow \mathbb{M}}{\Gamma \vdash \widehat{W}_{\mathbb{M}}(a, b) : \mathbb{M}} \mathbb{M}\text{in}_W \qquad \frac{\Gamma \vdash a : \mathbb{M} \quad \Gamma \vdash b : T_{\mathbb{M}}(a) \rightarrow \mathbb{M}}{\Gamma \vdash T_{\mathbb{M}}(\widehat{W}_{\mathbb{M}}(a, b)) = W_{(x:T_{\mathbb{M}}(a))} T_{\mathbb{M}}(b x)} \\[10pt]
\frac{\Gamma \vdash a : \mathbb{M} \quad \Gamma \vdash b : T_{\mathbb{M}}(a) \quad \Gamma \vdash c : T_{\mathbb{M}}(a)}{\Gamma \vdash \widehat{\text{Id}}_{\mathbb{M}}(a, b, c) : \mathbb{M}} \mathbb{M}\text{in}_{\text{Id}} \qquad \frac{\Gamma \vdash a : \mathbb{M} \quad \Gamma \vdash b : T_{\mathbb{M}}(a) \quad \Gamma \vdash c : T_{\mathbb{M}}(a)}{\Gamma \vdash T_{\mathbb{M}}(\widehat{\text{Id}}_{\mathbb{M}}(a, b, c)) = \text{Id}(T_{\mathbb{M}}(a), b, c)}
\end{array}$$

The rules concerning the codes $\widehat{U}_f$ for subuniverses of $\mathbb{M}$ are also included as follows.

$$\frac{\Gamma \vdash f : \Sigma_{(x:\mathbb{M})}(\mathbb{T}_{\mathbb{M}}(x) \rightarrow \mathbb{M}) \rightarrow \Sigma_{(x:\mathbb{M})}(\mathbb{T}_{\mathbb{M}}(x) \rightarrow \mathbb{M})}{\Gamma \vdash \widehat{U}_f : \mathbb{M}} \text{Min}_U$$

$$\frac{\Gamma \vdash f : \Sigma_{(x:\mathbb{M})}(\mathbb{T}_{\mathbb{M}}(x) \rightarrow \mathbb{M}) \rightarrow \Sigma_{(x:\mathbb{M})}(\mathbb{T}_{\mathbb{M}}(x) \rightarrow \mathbb{M})}{\Gamma \vdash \mathbb{T}_{\mathbb{M}}(\widehat{U}_f) = U_f} \text{T}_{\mathbb{M}\text{uni}}$$

$$\frac{\Gamma \vdash f : \Sigma_{(x:\mathbb{M})}(\mathbb{T}_{\mathbb{M}}(x) \rightarrow \mathbb{M}) \rightarrow \Sigma_{(x:\mathbb{M})}(\mathbb{T}_{\mathbb{M}}(x) \rightarrow \mathbb{M}) \quad \Gamma \vdash a : U_f}{\Gamma \vdash \widehat{T}_f(a) : \mathbb{M}} \text{Min}_{\widehat{T}}$$

As we have seen in Section 1, the Mahlo universe $\mathbb{M}$ reflects functions on families of small types in $\mathbb{M}$. The system **MLM** expresses this feature as follows. First of all, the Min_U rule states that there is a subuniverse U_f of $\mathbb{M}$ closed under $f : \Sigma_{(x:\mathbb{M})}(\mathbb{T}_{\mathbb{M}}(x) \rightarrow \mathbb{M}) \rightarrow \Sigma_{(x:\mathbb{M})}(\mathbb{T}_{\mathbb{M}}(x) \rightarrow \mathbb{M})$ with its code $\widehat{U}_f : \mathbb{M}$. The closure under f is then expressed by res_f^0 and res_f^1 , which provide the restriction of f to U_f : we have $\text{res}_f^0(a, b) : U_f$ and $\text{res}_f^1(a, b, c) : U_f$ for any $a : U_f$, $b : \mathbb{T}_f(a) \rightarrow U_f$ and $c : \mathbb{T}_f(\text{res}_f^0(a, b))$. The following computation rules given by the corresponding inference rules of **MLM** show that res_f^0 and res_f^1 actually provide the restriction of f to U_f via the “injection” $\widehat{T}_f$ from U_f to $\mathbb{M}$.

$$\widehat{T}_f(\text{res}_f^0(a, b)) = \mathbf{p}_1(f(\widehat{T}_f(a), \lambda x. \widehat{T}_f(b\ x))) : \mathbb{M}$$

$$\widehat{T}_f(\text{res}_f^1(a, b, c)) = \mathbf{p}_2(f(\widehat{T}_f(a), \lambda x. \widehat{T}_f(b\ x)))\ c : \mathbb{M}$$

That is, $\widehat{T}_f$ maps $\text{res}_f^0(a, b)$ to the left projection of $f(\widehat{T}_f(a), \lambda x. \widehat{T}_f(b\ x))$, and $\text{res}_f^1(a, b, c)$ to the right projection of $f(\widehat{T}_f(a), \lambda x. \widehat{T}_f(b\ x))$ applied to c .

We work informally in **MLM** to improve readability. For any context Γ , put

$$\text{Var}(\Gamma) := \{x \mid x : A \text{ belongs to } \Gamma \text{ for some type } A\}.$$

The Σ -type $\Sigma_{(x:\mathbb{M})}(\mathbb{T}_{\mathbb{M}}(x) \rightarrow \mathbb{M})$ is abbreviated as $\text{Fam}(\mathbb{M})$. The type constructor $+$ for sum types is treated as left associative, *i.e.*,

$$A_1 + A_2 + \cdots + A_n := (\cdots (A_1 + A_2) + \cdots + A_n).$$

Below we write $\text{Id}(A, a, b)$ as $a =_A b$.

Example 2.1 (Universes above Families of Small Types). The Mahlo universe $\mathbb{M}$ provides universes above families of small types as follows. Let $a_0 : \mathbb{M}$ and $b_0 : \mathbb{T}_{\mathbb{M}}(a_0) \rightarrow \mathbb{M}$ be given, and assume that neither a_0 nor b_0 has any free occurrence of a variable x . Defining $f_0 : \text{Fam}(\mathbb{M}) \rightarrow \text{Fam}(\mathbb{M})$ as $f_0 := \lambda x. (a_0, b_0)$, we have the universe U_{f_0} with the decoding function $\widehat{T}_{f_0}$. The universe U_{f_0} can be considered as a universe above (a_0, b_0) , as the following equations show:

$$\widehat{T}_{f_0}(\text{res}_{f_0}^0(\widehat{N}_{0f_0}, \lambda v. E_0(v))) = \mathbf{p}_1\left(f_0(\widehat{T}_{f_0}(\widehat{N}_{0f_0}), \lambda v. \widehat{T}_{f_0}(E_0(v)))\right) = \mathbf{p}_1((a_0, b_0)) = a_0$$

$$\widehat{T}_{f_0}(\text{res}_{f_0}^1(\widehat{N}_{0f_0}, \lambda v. E_0(v), c)) = \mathbf{p}_2\left(f_0(\widehat{T}_{f_0}(\widehat{N}_{0f_0}), \lambda v. \widehat{T}_{f_0}(E_0(v)))\right) c = \mathbf{p}_2((a_0, b_0))\ c = b_0\ c$$

That is, $\text{res}_{f_0}^0(\widehat{N}_{0f_0}, \lambda v. E_0(v))$ is a code for a_0 in U_{f_0} , and $\text{res}_{f_0}^1(\widehat{N}_{0f_0}, \lambda v. E_0(v), c)$ is a code for $b_0\ c$ in U_{f_0} , where $\widehat{N}_{0f_0}$ and $\lambda v. E_0(v)$ are dummy arguments.

Example 2.2 (Super Universe). A super universe, which is a universe closed under a universe operator, can be defined by reflecting a function with a parameter. Put $f_1 : \text{Fam}(\mathbb{M}) \rightarrow \text{Fam}(\mathbb{M})$ as $f_1 := \lambda z. y$, where the free variable (or the parameter) y is of type $\text{Fam}(\mathbb{M})$. We have $(\widehat{U}_{f_1}, \widehat{T}_{f_1}) : \text{Fam}(\mathbb{M})$ by reflecting f_1 . Then, we define $g := \lambda y. (\widehat{U}_{f_1}, \widehat{T}_{f_1})$, and obtain the universe U_g by reflecting g . Note that $g : \text{Fam}(\mathbb{M}) \rightarrow \text{Fam}(\mathbb{M})$ is a universe operator because it takes a family of small types in $\mathbb{M}$ and returns a universe above this family. The universe U_g is thus a super universe, because res_g^0 and res_g^1 show that U_g is actually closed under g as the equations below show: for any $a : U_g$ and $b : T_g(a) \rightarrow U_g$,

$$\begin{aligned} \widehat{T}_g(\text{res}_g^0(a, b)) &= \mathbf{p}_1(g(\widehat{T}_g(a), \lambda v. \widehat{T}_g(b v))) \\ &= \mathbf{p}_1(\widehat{U}_{\lambda z. (\widehat{T}_g(a), \lambda v. \widehat{T}_g(b v))}, \widehat{T}_{\lambda z. (\widehat{T}_g(a), \lambda v. \widehat{T}_g(b v))}) \\ &= \widehat{U}_{\lambda z. (\widehat{T}_g(a), \lambda v. \widehat{T}_g(b v))} \\ \widehat{T}_g(\text{res}_g^1(a, b, c)) &= \mathbf{p}_2(g(\widehat{T}_g(a), \lambda v. \widehat{T}_g(b v))) c \\ &= \mathbf{p}_2(\widehat{U}_{\lambda z. (\widehat{T}_g(a), \lambda v. \widehat{T}_g(b v))}, \widehat{T}_{\lambda z. (\widehat{T}_g(a), \lambda v. \widehat{T}_g(b v))}) c \\ &= \widehat{T}_{\lambda z. (\widehat{T}_g(a), \lambda v. \widehat{T}_g(b v))} c \end{aligned}$$

That is, $\text{res}_g^0(a, b)$ is a code of the universe above the family (a, b) , and $\text{res}_g^1(a, b)$ is a code of its decoding function. Notice that U_g also has a code $\text{res}_g^0(\text{res}_g^0(a, b), \text{res}_g^1(a, b))$ of the universe above the family $(\text{res}_g^0(a, b), \text{res}_g^1(a, b))$, and so on.

2.2. Construction of A Higher-Order Universe Operator. Following Palmgren [Pal98], we define the type O of *first-order operators* and the type $\text{Fam}(O)$ of *families of first-order operators* as

$$O := \text{Fam}(\mathbb{M}) \rightarrow \text{Fam}(\mathbb{M}), \quad \text{Fam}(O) := \Sigma_{(x:\mathbb{M})} (T_{\mathbb{M}}(x) \rightarrow O).$$

For instance, let f be of type $N_3 \rightarrow O$. Then, $(\widehat{N}_{3\mathbb{M}}, f) : \text{Fam}(O)$ is a family of first-order operators which consists of the operators $f \ 0_3, f \ 1_3, f \ 2_3 : O$. We note that O and $\text{Fam}(O)$ are just the lowest fragment of Palmgren's higher-order operators defined in [Pal98]: he introduced the notion of n -order operator for an arbitrary natural number n .

We define the higher-order operator $u^{\mathbb{M}} : \text{Fam}(O) \rightarrow O$ by using the reflection property of $\mathbb{M}$. Typically, this operator takes a family of universe operators as an input, and returns a universe operator which provides a universe closed under all of the operators in this family. So an output of $u^{\mathbb{M}}$ can be its input again, and this process will be iterated transfinitely to construct the type-theoretic counterparts of inaccessible sets.

Definition 2.3 (Operator $u^{\mathbb{M}}$, Link). The higher-order operator $u^{\mathbb{M}} : \text{Fam}(O) \rightarrow O$ is defined as follows. Let $(z, v) : \text{Fam}(O)$ and $(x, y) : \text{Fam}(\mathbb{M})$ be given. Then, by using $+\text{el}$ repeatedly, one can define

$$h^{\mathbb{M}} : \Pi_{(w:\text{Fam}(\mathbb{M}))} (N_1 + T_{\mathbb{M}}(x) + T_{\mathbb{M}}(z) + \Sigma_{(w':T_{\mathbb{M}}(z))} T_{\mathbb{M}}(\mathbf{p}_1(v \ w' \ w)) \rightarrow \mathbb{M})$$

such that

- $h^{\mathbb{M}} \ w \ (\text{iii } x_1) = x$ with $x_1 : N_1$,
- $h^{\mathbb{M}} \ w \ (\text{ijj } x_2) = y \ x_2$ with $x_2 : T_{\mathbb{M}}(x)$,

- $h^{\mathbb{M}} w \text{ (ij } y_1) = \mathbf{p}_1(v y_1 w)$ with $y_1 : T_{\mathbb{M}}(z)$,
- $h^{\mathbb{M}} w \text{ (j}(y_1, z_1)) = \mathbf{p}_2(v y_1 w) z_1$ with $y_1 : T_{\mathbb{M}}(z)$ and $z_1 : T_{\mathbb{M}}(\mathbf{p}_1(v y_1 w))$.

Next, put $f^{\mathbb{M}}[z, v, x, y] : \mathbf{O}$ as

$$f^{\mathbb{M}}[z, v, x, y] := \lambda w. (\widehat{N}_{1\mathbb{M}} \widehat{+}_{\mathbb{M}} x \widehat{+}_{\mathbb{M}} z \widehat{+}_{\mathbb{M}} \widehat{\Sigma}_{\mathbb{M}}(z, \lambda w'. \mathbf{p}_1(v w' w)), h^{\mathbb{M}} w).$$

Then, we define $u^{\mathbb{M}}(z, v)(x, y) := (\widehat{U}_{f^{\mathbb{M}}[z, v, x, y]}, \widehat{T}_{f^{\mathbb{M}}[z, v, x, y]})$. By applying Σel twice, we have $u^{\mathbb{M}} : \text{Fam}(\mathbf{O}) \rightarrow \mathbf{O}$.

Roughly speaking, $h^{\mathbb{M}} w$ in the definition above is a family of small types in $\mathbb{M}$ whose index set is

$$N_1 + T_{\mathbb{M}}(x) + T_{\mathbb{M}}(z) + \Sigma_{(w' : T_{\mathbb{M}}(z))} T_{\mathbb{M}}(\mathbf{p}_1(v w' w)).$$

That is,

- $h^{\mathbb{M}} w \text{ (iii } x_1)$ is the small type $x : \mathbb{M}$, where x_1 is of type N_1 ,
- $h^{\mathbb{M}} w \text{ (ij } x_2)$ is the small type $y x_2 : \mathbb{M}$ given by the family $y : T_{\mathbb{M}}(x) \rightarrow \mathbb{M}$ of small types and the index $x_2 : T_{\mathbb{M}}(x)$,
- $h^{\mathbb{M}} w \text{ (ij } y_1)$ is the small type $\mathbf{p}_1(v y_1 w) : \mathbb{M}$ given by the family $v : T_{\mathbb{M}}(z) \rightarrow \mathbf{O}$ of first-order operators, the index $y_1 : T_{\mathbb{M}}(z)$ and $w : \text{Fam}(\mathbb{M})$,
- $h^{\mathbb{M}} w \text{ (j}(y_1, z_1))$ is the small type $\mathbf{p}_2(v y_1 w) z_1 : \mathbb{M}$ given by the family $v : T_{\mathbb{M}}(z) \rightarrow \mathbf{O}$ of first-order operators, the index $y_1 : T_{\mathbb{M}}(z)$, $w : \text{Fam}(\mathbb{M})$ and $z_1 : T_{\mathbb{M}}(\mathbf{p}_1(v y_1 w))$.

In particular, the two small types $h^{\mathbb{M}} w \text{ (ij } y_1)$ and $h^{\mathbb{M}} w \text{ (j}(y_1, z_1))$ are the outputs of the first-order operator $v y_1$ from the family $v : T_{\mathbb{M}}(z) \rightarrow \mathbf{O}$.

For comparison, we show that super universes can be defined by using the operator $u^{\mathbb{M}}$ and this construction can be iterated. In [Pal98] Palmgren formulated the family of theories for higher-order universe operators, and showed that it provides not only a construction of super universes, but also the iteration of this construction. The difference between our example and Palmgren's consists in the use of the Mahlo universe $\mathbb{M}$: we defined the operator $u^{\mathbb{M}}$ by means of $\mathbb{M}$, which is absent in Palmgren's theories for higher-order universe operators. Our example explains how to iterate the super-universe construction in **MLM**.

Examples 2.4 (Super Universe and Universes above Families of Small Types). Take g as defined in Example 2.2, and let $(a, b) : \text{Fam}(\mathbb{M})$ be the case. Using the operator $u^{\mathbb{M}}$, we have the subuniverse of $\mathbb{M}$ and its decoding function

$$u^{\mathbb{M}}(\widehat{N}_{1\mathbb{M}}, \lambda x. \mathbf{E}_1(x, g))(a, b) = (\widehat{U}_{g^*}, \widehat{T}_{g^*}),$$

where we abbreviate $f^{\mathbb{M}}[\widehat{N}_{1\mathbb{M}}, \lambda x. \mathbf{E}_1(x, g), a, b]$ as g^* . This subuniverse has the codes for a and b c with $c : T_{\mathbb{M}}(a)$, as shown below.

$$\begin{aligned} \widehat{T}_{g^*}(\text{res}_{g^*}^1(\widehat{N}_{0g^*}, \lambda x. \mathbf{E}_0(x), \text{iii } 0_1)) &= h^{\mathbb{M}}(\widehat{N}_{0\mathbb{M}}, \lambda x. \widehat{T}_{g^*}(\mathbf{E}_0(x))) (\text{iii } 0_1) = a \\ \widehat{T}_{g^*}(\text{res}_{g^*}^1(\widehat{N}_{0g^*}, \lambda x. \mathbf{E}_0(x), \text{ij } c)) &= h^{\mathbb{M}}(\widehat{N}_{0\mathbb{M}}, \lambda x. \widehat{T}_{g^*}(\mathbf{E}_0(x))) (\text{ij } c) = b \end{aligned}$$

Moreover, the subuniverse $u^{\mathbb{M}}(\widehat{N}_{1\mathbb{M}}, \lambda x.E_1(x, g))(a, b)$ is closed under the universe operator g . Let $a' : U_{g^*}$ and $b' : T_{g^*}(a') \rightarrow U_{g^*}$ be given. We then have

$$\begin{aligned} \widehat{T}_{g^*}(\text{res}_{g^*}^1(a', b', \text{ij } 0_1)) &= h^{\mathbb{M}}(\widehat{T}_{g^*}(a'), \lambda x.\widehat{T}_{g^*}(b' x))(\text{ij } 0_1) \\ &= \mathbf{p}_1\left((\lambda y.E_1(y, g)) 0_1 (\widehat{T}_{g^*}(a'), \lambda x.\widehat{T}_{g^*}(b' x))\right) \\ &= \mathbf{p}_1\left(g(\widehat{T}_{g^*}(a'), \lambda x.\widehat{T}_{g^*}(b' x))\right) = \widehat{U}_{\lambda z.(\widehat{T}_{g^*}(a'), \lambda x.\widehat{T}_{g^*}(b' x))} \\ \widehat{T}_{g^*}(\text{res}_{g^*}^1(a', b', \text{j}(0_1, c))) &= h^{\mathbb{M}}(\widehat{T}_{g^*}(a'), \lambda x.\widehat{T}_{g^*}(b' x))(\text{j}(0_1, c)) \\ &= \mathbf{p}_2\left((\lambda y.E_1(y, g)) 0_1 (\widehat{T}_{g^*}(a'), \lambda x.\widehat{T}_{g^*}(b' x))\right) c \\ &= \mathbf{p}_2\left(g(\widehat{T}_{g^*}(a'), \lambda x.\widehat{T}_{g^*}(b' x))\right) c = \widehat{T}_{\lambda z.(\widehat{T}_{g^*}(a'), \lambda x.\widehat{T}_{g^*}(b' x))}(c) \end{aligned}$$

Thus, $\text{res}_{g^*}^1(a', b', \text{ij } 0_1)$ and $\lambda w'.\text{res}_{g^*}^1(a', b', \text{j}(0_1, w'))$ provide the codes in U_{g^*} for the universe obtained by applying the universe operator g to $\widehat{T}_{g^*}(a')$ and $\lambda x.\widehat{T}_{g^*}(b' x)$. This construction can be iterated as

$$\text{res}_{g^*}^1(\text{res}_{g^*}^1(a', b', \text{ij } 0_1), \lambda w'.\text{res}_{g^*}^1(a', b', \text{j}(0_1, w')), \text{ij } 0_1),$$

and so on.

Note that g^* is a first-order super-universe operator. So one can construct a universe closed under this super-universe operator by replacing g with g^* in the argument above, and this process can be iterated.

3. INACCESSIBLE SETS IN THE MAHLO UNIVERSE OF MLM

Below we first recall Aczel's constructive set theory **CZF** and then present Rathjen-Griffon-Palmgren's extension of **CZF** by inaccessible sets of all transfinite orders (Section 3.1). Next, we prove our main theorem that inaccessible sets of **CZF** _{π} are interpretable in **MLM** with the accessibility predicate *Acc* (Section 3.2).

3.1. CZF with Inaccessible Sets of All Transfinite Orders. The language of **CZF** is a usual first-order language with the equality $=$ and the membership relation $\in$ except that the restricted quantifiers $\forall x \in y$ and $\exists x \in y$ are primitive symbols. The symbol $\in$ is the only non-logical symbol in this language. The underlying logic of **CZF** is intuitionistic logic with equality, and the following set-theoretic axioms are postulated (*cf.* [AR01]).

Pairing:

$$\forall a \forall b \exists c \forall x (x \in c \leftrightarrow x = a \vee x = b)$$

Union:

$$\forall a \exists b \forall x (x \in b \leftrightarrow \exists y \in a (x \in y))$$

Infinity:

$$\exists a (\exists x (x \in a) \wedge \forall x \in a \exists y \in a (x \in y))$$

Extensionality:

$$\forall a \forall b (\forall x (x \in a \leftrightarrow x \in b) \rightarrow a = b)$$

Set Induction: For any formula $\varphi[x]$,

$$\forall a (\forall x \in a \varphi[x] \rightarrow \varphi[a]) \rightarrow \forall a \varphi[a].$$

Restricted Separation: For any restricted formula $\varphi[x]$,

$$\forall a \exists b \forall x (x \in b \leftrightarrow x \in a \wedge \varphi[x]),$$

where a formula is *restricted* if all of its quantifiers are restricted.

Subset Collection: For any formula $\psi[x, y, u]$,

$$\forall a \forall b \exists c \forall u$$

$$(\forall x \in a \exists y \in b \psi[x, y, u] \rightarrow \exists z \in c (\forall x \in a \exists y \in z \psi[x, y, u] \wedge \forall y \in z \exists x \in a \psi[x, y, u])).$$

Strong Collection: For any formula $\psi[x, y]$,

$$\forall a (\forall x \in a \exists y \psi[x, y] \rightarrow \exists b (\forall x \in a \exists y \in b \psi[x, y] \wedge \forall y \in b \exists x \in a \psi[x, y])).$$

Informally, one may suppose that the range of bound variables in **CZF** forms a collection of objects. We call these objects Aczel's *iterative sets* or *sets* shortly.

To state the axiom that there are inaccessible sets of all transfinite orders, we need to define the notion of regular set. For any set a , a is *transitive* if $\forall x \in a \forall y \in x (y \in a)$ holds.

Definition 3.1 (Regular Sets). A set a is regular if the following conditions are satisfied:

- (1) a is inhabited, i.e., $\exists y (y \in a)$,
- (2) a is transitive,
- (3) $\forall b \in a \forall R (R \subseteq b \times a \wedge \forall x \in b \exists y (\langle x, y \rangle \in R) \rightarrow \exists c \in a (\forall x \in b \exists y \in c (\langle x, y \rangle \in R) \wedge \forall y \in c \exists x \in b (\langle x, y \rangle \in R)))$.

The notion of inaccessible set is captured by *set-inaccessibility*.

Definition 3.2. A set a is set-inaccessible if the following conditions are satisfied:

- (1) a is regular,
- (2) $\forall x \in a \exists y \in a (x \subseteq y \wedge y \text{ is regular})$,
- (3) a is a model of **CZF**, i.e., the structure $\langle a, \in \upharpoonright (a \times a) \rangle$ is a model of **CZF**.

As in classical Zermelo-Fraenkel set theory, one can talk about *classes* in **CZF**. A class A is said to be *unbounded in* a class B if $\forall x \in B \exists y \in A (x \in y \wedge y \in B)$ holds. Let $\text{TC}(a)$ be the *transitive closure* of a set a with the following property (see [AR01, Proposition 4.2]):

$$\text{TC}(a) = a \cup \bigcup \{ \text{TC}(x) \mid x \in a \}.$$

Inaccessible sets of transfinite orders are defined along the hierarchy of Aczel's iterative sets.

Definition 3.3. Let a and b be sets, then b is a -set-inaccessible if b is set-inaccessible and there is a family $(x_c)_{c \in \text{TC}(a)}$ of sets such that for any $c \in \text{TC}(a)$,

- x_c is unbounded in b ,
- x_c consists of sets which are set-inaccessible,
- for any $y \in x_c$ and any $d \in \text{TC}(c)$, x_d is unbounded in y .

Informally speaking, a set a represents a transfinite “order” in the hierarchy of iterative sets, so a -set-inaccessibles can be considered as inaccessible sets of order a . Note that x_d in the third condition is well-defined because for any sets a, c and y , if $c \in \text{TC}(a)$ and $y \in \text{TC}(c)$ hold then $y \in \text{TC}(a)$ holds. This claim is proved by TC-induction (see [AR01, Proposition 4.3]) on a : if $c \in a$ holds, then we have $y \in \text{TC}(a)$ by $y \in \bigcup \{ \text{TC}(x) \mid x \in a \}$. If $c \in \bigcup \{ \text{TC}(x) \mid x \in a \}$ holds, then c is an element of $\text{TC}(x)$ for some $x \in \text{TC}(a)$. By the hypothesis of TC-induction, we have $y \in \text{TC}(x)$ and so $y \in \text{TC}(a)$ holds.

The first lemma below follows from the definition of a -set-inaccessibility, and the second lemma follows from the first (cf. [RGP98, Lemma 2.11]). For readers' convenience, we recall their proofs below.

Lemma 3.4 (CZF). *For any sets a, b and c , if a is b -set-inaccessible and $c \in \text{TC}(b)$ holds, then a is c -set-inaccessible.*

Proof. By assumption, there is a family $(x_d)_{d \in \text{TC}(b)}$ of sets witnessing the b -set-inaccessibility of a . To show its c -set-inaccessibility, we define a family $(y_d)_{d \in \text{TC}(c)}$ of sets as follows: for any $d \in \text{TC}(c)$, we have $d \in \text{TC}(b)$, so we put $y_d := x_d$. It immediately follows that y_d is unbounded in a and that y_d consists of sets which are set-inaccessible. Let $z \in y_d$ and $d' \in \text{TC}(d)$ be the case. It follows that $z \in x_d$ holds and so $x_{d'}$, namely, $y_{d'}$ is unbounded in z . $\square$

Lemma 3.5 (CZF). *Let a be set-inaccessible. Then, a is b -set-inaccessible if and only if for all $c \in b$, the class of c -set-inaccessibles is unbounded in a .*

Proof. Let a be b -set-inaccessible, so there is a family $(x_c)_{c \in \text{TC}(b)}$ witnessing the b -set-inaccessibility of a . We first show that for any $c \in \text{TC}(b)$, x_c consists of sets being c -set-inaccessible. Suppose that $c \in \text{TC}(b)$ holds and z is an element of x_c , then we verify that z is c -set-inaccessible. Since z is an element of x_c , z is set-inaccessible. Next, we define a family $(y_d)_{d \in \text{TC}(c)}$ of sets by putting $y_d := x_d$ for any $d \in \text{TC}(c) \subseteq \text{TC}(b)$. By definition, y_d consists of sets being set-inaccessible. We have $z \in x_c$ and $d \in \text{TC}(c)$, hence x_d is unbounded in z and so y_d is too. For any $v \in y_d$ and any $d' \in \text{TC}(d)$, we have $v \in x_d$ and so $x_{d'}$, i.e., $y_{d'}$ is unbounded in v . This concludes that x_c consists of sets which are c -set-inaccessible for any $c \in \text{TC}(b)$.

If $c \in b$ holds then $c \in \text{TC}(b)$ also holds. To show that the class of c -set-inaccessibles is unbounded in a , let $z \in a$ be the case. Since x_c is unbounded in a , there is a set $v \in x_c$ such that $z \in v$ and $v \in a$ hold. This implies that the class of c -set-inaccessibles is unbounded in a , because v is c -set-inaccessible.

Conversely, suppose that for all $c \in b$, the class of c -set-inaccessibles is unbounded in a . It follows from Lemma 3.4 that for all $d \in \text{TC}(b)$, the class of d -set-inaccessibles is unbounded in a : if $d \in b$ holds then the class of d -set-inaccessibles is unbounded in a by assumption. If $d \in \bigcup \{\text{TC}(x) \mid x \in b\}$ holds, then $d \in \text{TC}(x)$ holds and the class of x -set-inaccessibles is unbounded in a for some $x \in b$. Thus, for any $z \in a$, there is a set v such that v is x -set-inaccessible, z is an element of v and $v \in a$ holds. But v is also d -set-inaccessible by Lemma 3.4.

So we have $\forall d \in \text{TC}(b) \forall z \in a \exists v \in a (z \in v \wedge v \text{ is } d\text{-set-inaccessible})$, that is,

$$\begin{aligned} & \forall d \in \text{TC}(b) \forall z \in a \exists w \\ & \exists v \in a (w = \langle d, v, z \rangle \wedge z \in v \wedge v \text{ is } d\text{-set-inaccessible}). \end{aligned}$$

Apply **Strong Collection** twice to this statement, and take the union of the set whose existence follows from the two applications of **Strong Collection**. We then have a set S of triples $\langle d, v, z \rangle$ such that

- $z \in v \in a$ holds, $d \in \text{TC}(b)$ holds and v is d -set-inaccessible,
- for any $d \in \text{TC}(b)$ and any $z \in a$, there is a set $v \in a$ with $\langle d, v, z \rangle \in S$.

Using **Restricted Separation**, we define

$$S_d := \{v \in a \mid \exists z \in a \langle d, v, z \rangle \in S\}$$

for any $d \in \text{TC}(b)$. Then, it follows that

$$\forall d \in \text{TC}(b) \forall v \in S_d \text{ there exists a family } (x_{d'})_{d' \in \text{TC}(d)} \text{ of sets such that} \\ (x_{d'})_{d' \in \text{TC}(d)} \text{ witnesses the } d\text{-set-inaccessibility of } v.$$

Again, apply **Strong Collection** twice to this statement, and take the union of the resulting set. We then obtain a set $\mathcal{F}$ of families of sets such that

- for any $d \in \text{TC}(b)$ and any $v \in S_d$, there is a family $(x_{d'})_{d' \in \text{TC}(d)} \in \mathcal{F}$ witnessing the d -set-inaccessibility of v ,
- any $f \in \mathcal{F}$ is a family $(x_{d'})_{d' \in \text{TC}(d)}$ witnessing the d -set-inaccessibility of v for some $d \in \text{TC}(b)$ and some $v \in S_d$.

We verify that a is b -set-inaccessible. It follows from the assumption that a is set-inaccessible. We define a family $(y_d)_{d \in \text{TC}(b)}$ of sets: for any $d \in \text{TC}(b)$,

$$y_d := S_d \cup \bigcup \{x_d \mid (x_{d_2})_{d_2 \in \text{TC}(d_1)} \in \mathcal{F} \wedge d \in \text{TC}(d_1) \text{ for some } d_1 \in \text{TC}(b)\}.$$

By definition, any element of y_d is set-inaccessible. Moreover, y_d is unbounded in a because for any $z \in a$, there is a set $v \in a$ with $z \in v$ and $v \in S_d \subseteq y_d$.

Finally, we show that for any $u \in y_d$ and any $d' \in \text{TC}(d)$, $y_{d'}$ is unbounded in u . Let $s \in u$ be the case. If $u \in S_d$ holds, then there is a family $(x_{d'})_{d' \in \text{TC}(d)} \in \mathcal{F}$ witnessing the d -set-inaccessibility of u . Since $x_{d'}$ is unbounded in u , we have $s \in s' \in u$ for some $s' \in x_{d'} \subseteq y_{d'}$, hence $y_{d'}$ is unbounded in u . If u is an element of x_d such that $d \in \text{TC}(d_1)$ and $(x_{d_2})_{d_2 \in \text{TC}(d_1)} \in \mathcal{F}$ hold for some $d_1 \in \text{TC}(b)$, then $x_{d'}$ is unbounded in u because the family $(x_{d_2})_{d_2 \in \text{TC}(d_1)} \in \mathcal{F}$ witnesses the d_1 -set-inaccessibility of some set (cf. the last condition of Definition 3.3). So $s \in s' \in u$ holds for some $s' \in x_{d'} \subseteq y_{d'}$, hence $y_{d'}$ is unbounded in u also in this case. $\square$

The system $\mathbf{CZF}_\pi$ is obtained by adding the following axiom to $\mathbf{CZF}$.

Pi-Numbers:

$$\forall x \forall a \exists b (x \in b \wedge b \text{ is } a\text{-set-inaccessible}).$$

That is, $\mathbf{CZF}_\pi$ is the extension of $\mathbf{CZF}$ by the axiom that for any sets x and a , where a represents a transfinite “order” in the hierarchy of iterative sets, there is a set b which has x as its element and is a -set-inaccessible. This axiom says that for any set a , the a -set-inaccessibles are unbounded.

As mentioned in Section 1, Rathjen [Rat03] showed that $\mathbf{CZF}_\pi$ is interpretable in the system $\mathbf{CZFM}$, which is the extension of $\mathbf{CZF}$ obtained by adding the axiom asserting the existence of one Mahlo set to $\mathbf{CZF}$. In the remainder of this subsection, we briefly recall Rathjen’s result that $\mathbf{CZF}_\pi$ is interpretable in $\mathbf{CZFM}$, since this result is not explicitly stated in [Rat03].

Let $\mathbf{mv}^{(a,b)}$ be the class of all subsets R of $a \times b$ such that $\forall x \in a \exists y \in b (\langle x, y \rangle \in R)$ holds. The class $\mathbf{mv}^{(a,b)}$ is thus the class of all multi-valued functions from a to b . In [Rat03], the term “set-inaccessibles” means regular sets which are models of the fragment $\mathbf{CZF}^-$ of $\mathbf{CZF}$ obtained by omitting **Set Induction**. So we reformulate some definitions and lemmas in [Rat03] as follows, but the lemmas are provable exactly in the same way as [Rat03].

Definition 3.6 (Mahlo Sets). A set M is Mahlo if M is a regular set satisfying the following conditions:

- M is a model of $\mathbf{CZF}^-$, and

- for any $R \in \mathbf{mv}^M(M)$, there is a regular model $I \in M$ of $\mathbf{CZF}^-$ such that $\forall x \in I \exists y \in I \langle x, y \rangle \in R$ holds.

Lemma 3.7 ($\mathbf{CZF}^-$). *Let M be Mahlo. For any $R \in \mathbf{mv}^M(M)$ and any $a \in M$, there is a regular model $I \in M$ of $\mathbf{CZF}^-$ such that $a \in I$ and $\forall x \in I \exists y \in I \langle x, y \rangle \in R$ hold.*

Proof. Suppose that $R \in \mathbf{mv}^M(M)$ holds and a is an element of M . We define $S := \{\langle x, \langle a, y \rangle \rangle \mid \langle x, y \rangle \in R\}$. Then $S \in \mathbf{mv}^M(M)$ holds because M is closed under pairing, so there is a set $I \in M$ which is a regular model of $\mathbf{CZF}^-$, and satisfies $\forall x \in I \exists y \in I \langle x, y \rangle \in S$. Since S is regular, S is inhabited and so we have a set $x \in I$. We thus have $\langle x, \langle a, y \rangle \rangle \in S$ and $\langle a, y \rangle \in I$ for some y , hence $a \in I$ holds because I is transitive. Similarly, for any $x \in I$, there is an ordered pair $\langle a, y \rangle \in I$ with $\langle x, y \rangle \in R$, and we have $y \in I$ because I is transitive. $\square$

Lemma 3.8 ($\mathbf{CZF}^-$). *Let M be Mahlo. If $\forall x \in M \exists y \in M \varphi(x, y)$ holds, then there is a set $S \in \mathbf{mv}^M(M)$ satisfying $\forall x \forall y (\langle x, y \rangle \in S \rightarrow \varphi(x, y))$.*

Proof. Define $\psi(x, z) := \exists y \in M (z = \langle x, y \rangle \wedge \varphi(x, y))$, then, by assumption, we have $\forall x \in M \exists z \in M \psi(x, z)$ because M is closed under pairing. Using **Strong Collection**, there is a set S with $\forall x \in M \exists z \in S \psi(x, z)$ and $\forall z \in S \exists x \in M \psi(x, z)$. The property $\forall x \in M \exists z \in S \psi(x, z)$ implies that $S \in \mathbf{mv}^M(M)$ holds, while the property $\forall z \in S \exists x \in M \psi(x, z)$ implies that $\forall x \forall y (\langle x, y \rangle \in S \rightarrow \varphi(x, y))$ holds. $\square$

The following theorem is a straightforward adaptation of [Rat03, Theorem 4.10], and this theorem implies that a Mahlo set M models **Pi-Numbers**. Recall that a Mahlo set is a model of $\mathbf{CZF}^-$ by definition. Moreover, a Mahlo set also models **Set Induction**, because any Mahlo set is transitive and any transitive set models **Set Induction** in $\mathbf{CZF}$. It thus follows that $\mathbf{CZF}_\pi$ is interpretable in $\mathbf{CZFM}$: a Mahlo set M in $\mathbf{CZFM}$ models all axioms of $\mathbf{CZF}_\pi$.

Theorem 3.9 ($\mathbf{CZF}$). *Let M be Mahlo. For any set $a \in M$, the class of a -set-inaccessibles is unbounded in M .*

Proof. By induction on a . Let $x \in M$ be the case, and put a formula $\varphi(b, v)$ as

$$\varphi(b, v) := \exists z \in M (v = \langle b, z \rangle \wedge x \in z \wedge z \text{ is } b\text{-set-inaccessible}).$$

The following argument shows that $\forall b \in a \exists v \varphi(b, v)$ holds. Let b be a member of a , then we have

$$b \in M \rightarrow \text{the class of } b\text{-set-inaccessibles is unbounded in } M$$

by IH. We have $b \in M$ by the transitivity of M , so the class of b -set-inaccessibles is unbounded in M . This implies that $\varphi(b, v)$ holds for some v .

By **Strong Collection**, there is a set R satisfying the following:

$$\forall b \in a \exists v \in R \varphi(b, v) \wedge \forall v \in R \exists b \in a \varphi(b, v).$$

So R is a subset of $a \times M$, and $\forall b \in a \exists z (\langle b, z \rangle \in R)$ holds. Since M is regular, we have

$$\forall b \in a \exists z \in y (\langle b, z \rangle \in R)$$

for some $y \in M$. It then follows that for any $b \in a$, there exists a member z of y satisfying

- $\langle b, z \rangle = \langle b', z' \rangle$ holds,
- $x \in z'$ holds, and
- z' is b' -set-inaccessible

for some $b' \in a$ and $z' \in M$. But both $b = b'$ and $z = z'$ hold, so we have

$$\forall x \in M \exists y \in M \forall b \in a \exists z \in y (x \in z \wedge z \text{ is } b\text{-set-inaccessible}).$$

Lemmas 3.8 and 3.7 imply that for any $w \in M$, there exists a regular model $I \in M$ of $\mathbf{CZF}^-$ satisfying

$$w \in I \wedge \forall x \in I \exists y \in I \forall b \in a \exists z \in y (x \in z \wedge z \text{ is } b\text{-set-inaccessible}).$$

By the transitivity of I , we have

$$\forall x \in I \forall b \in a \exists z \in I (x \in z \wedge z \text{ is } b\text{-set-inaccessible}),$$

so the class of b -set-inaccessibles is unbounded in I for any $b \in a$. By the transitivity of I again, I satisfies **Set Induction**, hence I is also a model of $\mathbf{CZF}$. Moreover, one can see by definition that any b -set-inaccessible is regular, so the class of regular sets is unbounded in I . Therefore, I is set-inaccessible.

It follows by Lemma 3.5 that I is a -set-inaccessible, therefore a -set-inaccessibles are unbounded in M . $\square$

3.2. Construction of Inaccessible Sets in Mahlo Universe. To show that $\mathbf{CZF}_\pi$ is interpretable in $\mathbf{MLM}$ with the accessibility predicate Acc , we recall Aczel's interpretation of $\mathbf{CZF}$ in $\mathbf{MLTT}$.

First of all, we define the type-theoretic counterpart of Aczel's iterative sets. Let a W -type $W_{(x:A)}B$ be given. It is straightforward to define the functions $\text{index} : W_{(x:A)}B \rightarrow A$ and $\text{pred} : \Pi_{(y:W_{(x:A)}B)}B[\text{index } y/x] \rightarrow W_{(x:A)}B$ satisfying

$$\text{index } \sup(a, b) = a, \quad \text{pred } \sup(a, b) = b.$$

The W -type $\mathbb{V}$, which is the type of iterative sets in Aczel's interpretation of $\mathbf{CZF}$, is defined as

$$\mathbb{V} := W_{(x:\mathbb{M})}T_{\mathbb{M}}(x).$$

We call terms of type $\mathbb{V}$ in some contexts *sets*, and denote them by lowercase Greek letters such as α, β, γ possibly with suffixes. For any $\alpha : \mathbb{V}$, we write $T_{\mathbb{M}}(\text{index } \alpha)$ as $\bar{\alpha}$, following [RGP98]. Moreover, we adopt the abbreviations below: for any term F such that $\Gamma, \beta : \mathbb{V} \vdash F$ type holds for some context Γ ,

$$\forall \beta \in \alpha F := \Pi_{(x:\bar{\alpha})}F[\text{pred } \alpha x/\beta], \quad \exists \beta \in \alpha F := \Sigma_{(x:\bar{\alpha})}F[\text{pred } \alpha x/\beta].$$

Note that the elimination rule for $\mathbb{V} = W_{(x:\mathbb{M})}T_{\mathbb{M}}(x)$ provides the transfinite induction principle on $\alpha : \mathbb{V}$. That is, we have a term

$$\text{TI}_F : \Pi_{(\alpha:\mathbb{V})}(\forall \gamma \in \alpha F[\gamma/\beta] \rightarrow F[\alpha/\beta]) \rightarrow \Pi_{(\alpha:\mathbb{V})}F[\alpha/\beta],$$

where $\Gamma, \beta : \mathbb{V} \vdash F$ type holds for some context Γ .

Next, we define the interpretation of the equality $a = b$ and the membership relation $a \in b$ in $\mathbf{CZF}$ as an equivalence relation $\doteq$ and a binary relation $\in$ on $\mathbb{V}$, respectively. The relation $\doteq$ on $\mathbb{V}$ is defined by applying the elimination rule for W -types twice:

$$\sup(a, f) \doteq \sup(b, g) := (\Pi_{(x:T_{\mathbb{M}}(a))}\Sigma_{(y:T_{\mathbb{M}}(b))}f x \doteq g y) \times (\Pi_{(y:T_{\mathbb{M}}(b))}\Sigma_{(x:T_{\mathbb{M}}(a))}f x \doteq g y).$$

For any $\alpha, \beta : \mathbb{V}$, we define $\alpha \in \beta$ as $\alpha \in \beta := \exists \gamma \in \beta(\alpha \doteq \gamma)$. Below we use the following facts about these relations $\doteq$ and $\in$ without mentioning them. Note that by Lemma 3.10.(1) below we have $\alpha \doteq \sup(\text{index } \alpha, \text{pred } \alpha)$ for any $\alpha : \mathbb{V}$.

Lemma 3.10 (Link). *The following holds:*

- (1) *For any $\alpha : \mathbb{V}$, we have $\alpha =_{\mathbb{V}} \text{sup}(\text{index } \alpha, \text{pred } \alpha)$.*
- (2) *For any $\alpha, \beta : \mathbb{V}$, if $\alpha \doteq \beta$ holds then $\Pi_{(\gamma:\mathbb{V})}(\gamma \in \alpha \rightarrow \gamma \in \beta)$ holds.*
- (3) *For any $a : \mathbb{M}$ and $f, g : \mathbb{T}_{\mathbb{M}}a \rightarrow \mathbb{V}$, if $\Pi_{(x:\mathbb{T}_{\mathbb{M}}a)}f x \doteq g x$ holds then $\text{sup}(a, f) \doteq \text{sup}(a, g)$ holds.*
- (4) *For any $\alpha, \beta : \mathbb{V}$, if both $\Pi_{(\gamma:\mathbb{V})}(\gamma \in \alpha \rightarrow \gamma \in \beta)$ and $\Pi_{(\gamma:\mathbb{V})}(\gamma \in \beta \rightarrow \gamma \in \alpha)$ holds then $\alpha \doteq \beta$ holds.*

Proof. The statements (1) and (2) are provable by the induction principle of $\mathbb{V}$. The statement (3) follows straightforwardly, and (4) holds because the axiom **Extensionality** is interpretable in **MLTT**. $\square$

We interpret all formulas of **CZF** in **MLTT** by using these relations $\doteq$ and $\in$ on $\mathbb{V}$. The formulas $x = y$ and $x \in y$ in the language of **CZF** are interpreted as the relations $x \doteq y$ and $x \in y$ on $\mathbb{V}$, respectively. Compound formulas of **CZF** are interpreted as follows:

$$\begin{aligned}
 (\varphi \wedge \psi)^* &:= \varphi^* \times \psi^* & (\varphi \vee \psi)^* &:= \varphi^* + \psi^* \\
 (\varphi \rightarrow \psi)^* &:= \varphi^* \rightarrow \psi^* & \perp^* &:= \mathbb{N}_0 \\
 (\forall x \in y \varphi)^* &:= \forall x \in y \varphi^* & (\exists x \in y \varphi)^* &:= \exists x \in y \varphi^* \\
 (\forall x \varphi)^* &:= \Pi_{(\alpha:\mathbb{V})} \varphi^* & (\exists x \varphi)^* &:= \Sigma_{(\alpha:\mathbb{V})} \varphi^*
 \end{aligned}$$

It is then straightforward to show by induction on the length of a given **CZF**-formula φ that if a list $x_1, \dots, x_n$ of variables contains all free variables in φ then the judgement $x_1 : \mathbb{V}, \dots, x_n : \mathbb{V} \vdash \varphi^*$ type is derivable in **MLTT**. Moreover, as suggested in [Acz82, Remarks 3.5], the following useful lemma is provable by induction on the length of a given **CZF**-formula.

Lemma 3.11. *Let $\varphi[x_1, \dots, x_n, x]$ be an arbitrary formula of **CZF** such that $x_1, \dots, x_n$ and x contain all free variable occurring in φ . Then, the statement*

$$\Pi_{(x_1:\mathbb{V})} \cdots \Pi_{(x_n:\mathbb{V})} \Pi_{(y:\mathbb{V})} \Pi_{(z:\mathbb{V})} (y \doteq z \rightarrow \varphi^*[x_1, \dots, x_n, y] \rightarrow \varphi^*[x_1, \dots, x_n, z])$$

*is derivable in **MLTT**.*

In **MLTT**, one is able to define the corresponding operators to the set-theoretic axioms of **CZF** (see [Acz78, Acz82]). For instance, the union operator and the binary union operator for elements of $\mathbb{V}$ can be defined: let $A \leftrightarrow B$ be an abbreviation of $(A \rightarrow B) \times (B \rightarrow A)$, then the type-theoretic axiom of choice in **MLTT** provides the two terms $\bigcup : \mathbb{V} \rightarrow \mathbb{V}$ and $\bigcup : \mathbb{V} \rightarrow \mathbb{V} \rightarrow \mathbb{V}$ satisfying

$$\begin{aligned}
 \Pi_{(\alpha:\mathbb{V})} \Pi_{(\gamma:\mathbb{V})} (\gamma \in \bigcup \alpha \leftrightarrow \exists \delta \in \alpha (\gamma \in \delta)), \\
 \Pi_{(\alpha_1:\mathbb{V})} \Pi_{(\alpha_2:\mathbb{V})} \Pi_{(\gamma:\mathbb{V})} (\gamma \in \alpha_1 \cup \alpha_2 \leftrightarrow (\gamma \in \alpha_1) + (\gamma \in \alpha_2)).
 \end{aligned}$$

The term $\bigcup$ enables to interpret the axiom **Union** in **MLTT**, that is, one can show that the interpretation of the axiom **Union** is derivable in **MLTT**. In a similar manner, one can verify that all axioms of **CZF** are interpretable in **MLTT** (see [Acz78, Acz82] for details).

For our purpose, the axiom **Pi-Numbers** remains to be interpreted in **MLM**. Below we first define the transitive closure function tc on $\mathbb{V}$ to provide an interpretation of the set-theoretic transitive closure. Next, we define the accessibility predicate Acc on $\mathbb{V}$ and

construct the type-theoretic counterparts $\mathbf{V}_{(a,f)}^{\alpha,t}$ of inaccessible sets by iterating the higher-order universe operator $u^{\mathbb{M}}$ along the accessibility given by Acc . Finally, it is shown that inaccessible sets of all transfinite orders in **CZF** are indeed interpreted by these counterparts.

Following [RGP98], we define the transitive closure function on $\mathbb{V}$ as follows.

Definition 3.12 (Transitive Closure, **Link**). The transitive closure function $\text{tc} : \mathbb{V} \rightarrow \mathbb{V}$ is defined as

$$\text{tc}(\text{sup}(a, f)) := \text{sup}(a, f) \cup (\bigcup \text{sup}(a, \lambda x. \text{tc}(f x))).$$

We put $\alpha_{\text{tc}} := \text{tc}(\alpha)$ as in [RGP98].

Note that, by definition, $\overline{\text{sup}(a, f)_{\text{tc}}}$ is definitionally equal to $T_{\mathbb{M}}(a) + \Sigma_{(z:T_{\mathbb{M}}(a))}(\overline{f z})_{\text{tc}}$. In addition, we define the accessibility predicate on $\mathbb{V}$ as an inductive predicate. This predicate is used to iterate the higher-order operator $u^{\mathbb{M}}$ along a given $\alpha : \mathbb{V}$ (cf. Definition 3.14).

Definition 3.13 (Accessibility Predicate, **Link**). The predicate Acc on $\mathbb{V}$ is defined inductively: the formation, introduction, elimination and computation rules are as follows.

$$\begin{array}{c} \frac{\Gamma \vdash \alpha : \mathbb{V}}{\Gamma \vdash \text{Acc } \alpha \text{ type}} \text{ fm} \quad \frac{\Gamma \vdash \alpha : \mathbb{V} \quad \Gamma \vdash f : \Pi_{(x:\overline{\alpha_{\text{tc}}})} \text{Acc}(\text{pred } \alpha_{\text{tc}} x)}{\Gamma \vdash \text{prog } f : \text{Acc } \alpha} \text{ in} \\ \\ \frac{\begin{array}{c} \Gamma \vdash \alpha : \mathbb{V} \\ \Gamma \vdash t : \text{Acc } \alpha \\ \Gamma, \beta : \mathbb{V}, x : \text{Acc } \beta \vdash C \text{ type} \end{array} \quad \Gamma \vdash g : \Pi_{(\gamma:\mathbb{V})} \Pi_{(f:\Pi_{(y:\overline{\gamma_{\text{tc}}})} \text{Acc}(\text{pred } \gamma_{\text{tc}} y))} ((\Pi_{(y:\overline{\gamma_{\text{tc}}})} C[\text{pred } \gamma_{\text{tc}} y/\beta, f y/x]) \rightarrow C[\gamma/\beta, \text{prog } f/x])}{\Gamma \vdash E_{\text{Acc}}(\alpha, t, g) : C[\alpha/\beta, t/x]} \text{ el} \\ \\ \frac{\begin{array}{c} \Gamma \vdash \alpha : \mathbb{V} \\ \Gamma \vdash f : \Pi_{(y:\overline{\alpha_{\text{tc}}})} \text{Acc}(\text{pred } \alpha_{\text{tc}} y) \\ \Gamma, \beta : \mathbb{V}, x : \text{Acc } \beta \vdash C \text{ type} \end{array} \quad \Gamma \vdash g : \Pi_{(\gamma:\mathbb{V})} \Pi_{(f:\Pi_{(y:\overline{\gamma_{\text{tc}}})} \text{Acc}(\text{pred } \gamma_{\text{tc}} y))} ((\Pi_{(y:\overline{\gamma_{\text{tc}}})} C[\text{pred } \gamma_{\text{tc}} y/\beta, f y/x]) \rightarrow C[\gamma/\beta, \text{prog } f/x])}{\Gamma \vdash E_{\text{Acc}}(\alpha, \text{prog } f, g) = g \alpha f (\lambda y. E_{\text{Acc}}(\text{pred } \alpha_{\text{tc}} y, f y, g)) : C[\alpha/\beta, \text{prog } f/x]} \text{ eq} \end{array}$$

Informally speaking, the family $\bigcup_{x:\overline{\alpha_{\text{tc}}}} \text{pred } \alpha_{\text{tc}} x$ of sets exhausts all sets which are needed to construct the set α , and the introduction rule of Acc means that α is accessible if any of such sets is accessible. The elimination rule of Acc enables to provide an inductive argument for $C[\gamma/\beta, \text{prog } f/x]$ with the induction hypothesis that $C[\text{pred } \gamma_{\text{tc}} y/\beta, f y/x]$ holds for any $y : \overline{\gamma_{\text{tc}}}$.

We denote by $\mathbf{MLM}_{\text{acc}}$ the system obtained by extending **MLM** with the rules for Acc . In the remainder of this section we work in $\mathbf{MLM}_{\text{acc}}$. The induction principle on Acc provides the iterator of the operator $u^{\mathbb{M}}$ below.

Definition 3.14 (Iterator of $u^{\mathbb{M}}$ along Accessibility, **Link**). We define $\Phi : \Pi_{(\alpha:\mathbb{V})} (\text{Acc } \alpha \rightarrow \mathcal{O})$ by induction on $t : \text{Acc } \alpha$:

$$\Phi \alpha (\text{prog } f) (c, h) := u^{\mathbb{M}}(\text{index } \alpha_{\text{tc}}, \lambda x. \Phi(\text{pred } \alpha_{\text{tc}} x) (f x)) (c, h).$$

We abbreviate $\Phi \alpha t (c, h)$ as $\Phi_{(c,h)}^{\alpha,t}$.

Here the iterator Φ takes a set α and a proof $\mathbf{prog} f$ of its accessibility with a family $(c, h) : \mathbf{Fam}(\mathbb{M})$ and returns the subuniverse U of $\mathbb{M}$ such that

- U is closed under the universe operator $\Phi(\mathbf{pred} \alpha_{tc} x)(f x)$ for any $x : \overline{\alpha_{tc}}$, which is obtained by the induction hypothesis, and
- U includes the codes of c and h .

The hierarchy of such subuniverses is defined by applying the left and right projection to the output of the iterator Φ .

Definition 3.15 (Transfinite Hierarchy of Subuniverses, **Link**). Let $\alpha : \mathbb{V}$, $t : \mathbf{Acc} \alpha$ and $(a, f) : \mathbf{Fam}(\mathbb{M})$ be given. We define the transfinite hierarchy of subuniverses with their decoding functions $\hat{T}_{(a,f)}^{\alpha,t}$ along $\mathbf{Acc}$.

$$\begin{aligned} \hat{\mathcal{U}}_{(a,f)}^{\alpha,t} &:= \mathbf{p}_1(\Phi_{(a,f)}^{\alpha,t}) : \mathbb{M} & \mathcal{U}_{(a,f)}^{\alpha,t} &:= \mathbf{T}_{\mathbb{M}}(\hat{\mathcal{U}}_{(a,f)}^{\alpha,t}) \\ \hat{T}_{(a,f)}^{\alpha,t} &:= \mathbf{p}_2(\Phi_{(a,f)}^{\alpha,t}) : \mathcal{U}_{(a,f)}^{\alpha,t} \rightarrow \mathbb{M} & T_{(a,f)}^{\alpha,t} &:= \lambda x. \mathbf{T}_{\mathbb{M}}(\hat{T}_{(a,f)}^{\alpha,t}(x)) \end{aligned}$$

Each of subuniverses $\mathcal{U}_{(a,f)}^{\alpha,t}$ provides the type $\mathcal{V}_{(a,f)}^{\alpha,t}$ of iterative sets on it, as we will define below. The transfinite hierarchy of iterative sets consists of the sets each of which is obtained by embedding the sets in $\mathcal{V}_{(a,f)}^{\alpha,t}$ into $\mathbb{V}$ via the mapping $\mathbf{h}_{(a,f)}^{\alpha,t}$ defined below. One can find the Agda code for the definition of this hierarchy of iterative sets in `IterativeSetHierarchy.agda` of [Tak25], while the definition of the type $\mathcal{V}_{(a,f)}^{\alpha,t}$ is formalised in `TypeHierarchy.agda`.

Definition 3.16 (Transfinite Hierarchy of Iterative Sets). The transfinite hierarchy of iterative sets is defined via the type $\mathcal{V}_{(a,f)}^{\alpha,t}$ and the mapping $\mathbf{h}_{(a,f)}^{\alpha,t} : \mathcal{V}_{(a,f)}^{\alpha,t} \rightarrow \mathbb{V}$:

$$\begin{aligned} \hat{\mathcal{V}}_{(a,f)}^{\alpha,t} &:= \widehat{\mathbf{W}}_{\mathbb{M}}(\hat{\mathcal{U}}_{(a,f)}^{\alpha,t}, \hat{T}_{(a,f)}^{\alpha,t}) : \mathbb{M} \\ \mathcal{V}_{(a,f)}^{\alpha,t} &:= \mathbf{T}_{\mathbb{M}}(\hat{\mathcal{V}}_{(a,f)}^{\alpha,t}) \\ \mathbf{h}_{(a,f)}^{\alpha,t} \mathbf{sup}(b, g) &:= \mathbf{sup}(\hat{T}_{(a,f)}^{\alpha,t}(b), \lambda x. \mathbf{h}_{(a,f)}^{\alpha,t}(g x)) \\ \mathbf{V}_{(a,f)}^{\alpha,t} &:= \mathbf{sup}(\hat{\mathcal{V}}_{(a,f)}^{\alpha,t}, \mathbf{h}_{(a,f)}^{\alpha,t}) : \mathbb{V} \end{aligned}$$

Lemma 3.21 below shows that $\mathbf{V}_{(a,f)}^{\alpha,t}$ is α -set-inaccessible: it is the hierarchy defined above that validates the axiom **Pi-Numbers** in $\mathbf{MLM}_{\mathbf{acc}}$. The set-theoretic terms “transitive”, “regular”, “ α -set-inaccessible” and “unbounded” below are in fact their translations in **MLTT**.

In a way similar to define the relations $\dot{=}$ and $\in$ on $\mathbb{V}$, the equivalence relation $\dot{=}_{(a,f)}^{\alpha,t}$ and the membership relation $\in_{(a,f)}^{\alpha,t}$ on $\mathcal{V}_{(a,f)}^{\alpha,t}$ can be defined, as remarked in [RGP98]. One can prove the following lemma for these relations. The Agda code for the proofs of the statements (1), (3) and (4) can be found in `TypeHierarchy.agda` of [Tak25]. On the other hand, the code for the proofs of statements (2) and (5) can be found in `IterativeSetHierarchy.agda` and `Inaccessibles/Basics.agda`, respectively. Note that the Agda code for the statement (5) depends on a postulate on which we will comment at the end of this section.

Lemma 3.17. *The following holds:*

- (1) For any $x : \mathcal{V}_{(a,f)}^{\alpha,t}$, we have $x \dot{=}_{(a,f)}^{\alpha,t} \mathbf{sup}(\mathbf{index} x, \mathbf{pred} x)$.
- (2) $\mathbf{V}_{(a,f)}^{\alpha,t}$ is transitive.

- (3) For any $x, y : \mathcal{V}_{(a,f)}^{\alpha,t}$, $x \in_{(a,f)}^{\alpha,t} y$ holds if and only if $\mathbf{h}_{(a,f)}^{\alpha,t} x \in \mathbf{h}_{(a,f)}^{\alpha,t} y$ holds.
- (4) For any $x, y : \mathcal{V}_{(a,f)}^{\alpha,t}$, $x \dot{=}^{\alpha,t}_{(a,f)} y$ holds if and only if $\mathbf{h}_{(a,f)}^{\alpha,t} x \dot{=}^{\alpha,t}_{(a,f)} \mathbf{h}_{(a,f)}^{\alpha,t} y$ holds.
- (5) $\mathbf{V}_{(a,f)}^{\alpha,t}$ is set-inaccessible.

Proof. The statement (1) is proved by induction on $\mathcal{V}_{(a,f)}^{\alpha,t}$. For the proofs of (2), (3) and (4), see [RGP98, Lemma 4.7]; for the proof of (5), see [RGP98, Corollary 4.8]. Note that the definition of transfinite hierarchy of iterative sets in [RGP98] does not use the accessibility predicate Acc . Moreover, the proofs in [RGP98] do not assume much more than that each type $\mathcal{V}_{(a,f)}^{\alpha,t}$ is an instance of Aczel's type of iterative sets (cf. [Acz82, § 1.9] and [Acz86, § 1]). $\square$

By a standard argument using the induction principle of identity types, we have the so-called indiscernibility of identicals, which is also called the transport lemma: for any A, B with $\Gamma, z : A \vdash B$ type,

$$\text{transport}^B : \Pi_{(x:A)} \Pi_{(y:A)} (x =_A y \rightarrow B[x/z] \rightarrow B[y/z]).$$

For the sake of brevity, we often treat the first two arguments of transport^B as implicit.

Lemma 3.18 (Link). *For any $a, b : \mathbb{M}$, if $p : a =_{\mathbb{M}} b$ holds then for any $g : T_{\mathbb{M}}(b) \rightarrow \mathbb{V}$,*

$$\sup(a, \lambda x. g(\text{transport}^{T_{\mathbb{M}}(z)} a b p x)) =_{\mathbb{V}} \sup(b, g)$$

holds.

Proof. One can prove the lemma by the induction principle of identity types: it suffices to verify that the claim

$$\sup(a, \lambda x. g(\text{transport}^{T_{\mathbb{M}}(z)} a a \text{refl}(a) x)) =_{\mathbb{V}} \sup(a, g)$$

holds. Since $\text{transport}^{T_{\mathbb{M}}(z)} a a \text{refl}(a)$ is definitionally equal to the identity function on $T_{\mathbb{M}}(a)$, the claim holds by the reflexivity of identity types and the η -rule for Π -types. $\square$

We prove one more auxiliary lemma below.

Lemma 3.19 (Link). *For any $\alpha : \mathbb{V}$, $t : \text{Acc } \alpha$, $a : \mathbb{M}$, $g : T_{\mathbb{M}}(a) \rightarrow \mathbb{M}$ and $x : T_{\mathbb{M}}(a)$, there is a term c of type $\mathcal{U}_{(a,g)}^{\alpha,t}$ such that $\widehat{T}_{(a,g)}^{\alpha,t}(c) =_{\mathbb{M}} g x$ holds.*

Proof. We prove the lemma by transfinite induction on $t : \text{Acc } \alpha$, so let t be $\text{prog } f'$. We have

$$\begin{aligned} \mathcal{U}_{(a,g)}^{\alpha, \text{prog } f'} &= T_{\mathbb{M}}(\mathbf{p}_1(\Phi \alpha (\text{prog } f') (a, g))) \\ &= T_{\mathbb{M}}(\mathbf{p}_1(\mathbf{u}^{\mathbb{M}}(\text{index } \alpha_{\text{tc}}, \lambda y. \Phi (\text{pred } \alpha_{\text{tc}} y) (f' y)) (a, g))) \\ &= \mathbf{U}_{f^{\mathbb{M}}[\text{index } \alpha_{\text{tc}}, \lambda y. \Phi (\text{pred } \alpha_{\text{tc}} y) (f' y), a, g]}. \end{aligned}$$

Put $f^* := f^{\mathbb{M}}[\text{index } \alpha_{\text{tc}}, \lambda y. \Phi (\text{pred } \alpha_{\text{tc}} y) (f' y), a, g]$. Note that we have $\mathbf{U}_{f^*} = \mathcal{U}_{(a,g)}^{\alpha, \text{prog } f'}$ and $\widehat{T}_{f^*} = \widehat{T}_{(a,g)}^{\alpha, \text{prog } f'}$. It then follows that

$$\widehat{T}_{f^*}(\text{res}_{f^*}^1(\widehat{N}_{0f^*}, \lambda v. \mathbf{E}_0 v, \text{ij } x)) = \mathbf{p}_2(f^*(\widehat{T}_{f^*}(\widehat{N}_{0f^*}), \lambda v. \widehat{T}_{f^*}(\mathbf{E}_0 v), \text{ij } x)) = g x$$

holds. So the term $\text{res}_{f^*}^1(\widehat{N}_{0f^*}, \lambda v. \mathbf{E}_0 v, \text{ij } x) : \mathbf{U}_{f^*}$ satisfies the claim. $\square$

The following is the main lemma in our proof. Informally, this lemma says that $\mathbf{V}_{(a,f)}^{\alpha, \text{prog } f'}$ is unbounded in $(\text{pred } \alpha_{\text{tc}} u)$ -set-inaccessibles for any $u : \overline{\alpha_{\text{tc}}}$.

Lemma 3.20 (Main Lemma, Link). *If $\xi \in \mathbf{V}_{(a,f)}^{\alpha, \text{prog } f'}$ holds, then for any $u : \overline{\alpha_{\text{tc}}}$, there are terms $b : \mathcal{U}_{(a,f)}^{\alpha, \text{prog } f'}$ and $g : \mathbf{T}_{(a,f)}^{\alpha, \text{prog } f'}(b) \rightarrow \mathcal{U}_{(a,f)}^{\alpha, \text{prog } f'}$ such that*

$$\xi \in \mathbf{V}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u} \in \mathbf{V}_{(a,f)}^{\alpha, \text{prog } f'}$$

holds with $b^ := \widehat{\mathbf{T}}_{(a,f)}^{\alpha, \text{prog } f'}(b)$ and $g^* := \lambda x. \widehat{\mathbf{T}}_{(a,f)}^{\alpha, \text{prog } f'}(g x)$.*

Proof. Below we abbreviate $\text{prog } f'$ as t . Since $\mathbf{V}_{(a,f)}^{\alpha, t}$ is set-inaccessible by Lemma 3.17.(5), the singleton $\{\xi\}$ also belongs to $\mathbf{V}_{(a,f)}^{\alpha, t}$. Moreover, there is a regular set $\beta \in \mathbf{V}_{(a,f)}^{\alpha, t}$ such that $\xi \in \beta$ holds and β is transitive. It follows that $\beta \doteq \mathbf{h}_{(a,f)}^{\alpha, t} x$ holds for some $x : \mathcal{V}_{(a,f)}^{\alpha, t}$. We have $x \doteq_{(a,f)}^{\alpha, t} \text{sup}(\text{index } x, \text{pred } x)$ by Lemma 3.17.(1), so

$$\beta \doteq \mathbf{h}_{(a,f)}^{\alpha, t} x \doteq \mathbf{h}_{(a,f)}^{\alpha, t} \text{sup}(\text{index } x, \text{pred } x) = \text{sup}(\widehat{\mathbf{T}}_{(a,f)}^{\alpha, t}(\text{index } x), \lambda y. \mathbf{h}_{(a,f)}^{\alpha, t}(\text{pred } x y))$$

holds by Lemma 3.17.(4).

Put

- $b := \text{index } x : \mathcal{U}_{(a,f)}^{\alpha, t}$ and $g := \lambda z. \text{index}(\text{pred } x z) : \mathbf{T}_{(a,f)}^{\alpha, t}(b) \rightarrow \mathcal{U}_{(a,f)}^{\alpha, t}$, where $\mathbf{T}_{(a,f)}^{\alpha, t}(b) = \mathbf{T}_{\mathbb{M}}(\widehat{\mathbf{T}}_{(a,f)}^{\alpha, t}(b))$ holds by definition,
- $b^* := \widehat{\mathbf{T}}_{(a,f)}^{\alpha, t}(b) : \mathbb{M}$ and $g^* := \lambda w. \widehat{\mathbf{T}}_{(a,f)}^{\alpha, t}(g w) : \mathbf{T}_{(a,f)}^{\alpha, t}(b) \rightarrow \mathbb{M}$,
- $\widehat{\mathcal{U}} := \widehat{\mathcal{U}}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u}$.

We first show that $\mathcal{U}_{(a,f)}^{\alpha, t}$ has a code for $\widehat{\mathcal{U}}$. We have by definition

$$\begin{aligned} \mathcal{U}_{(a,f)}^{\alpha, t} &= \mathbf{T}_{\mathbb{M}}(\mathbf{p}_1(\Phi_{(a,f)}^{\alpha, t})) \\ &= \mathbf{T}_{\mathbb{M}}(\mathbf{p}_1(\mathbf{u}^{\mathbb{M}}(\text{index } \alpha_{\text{tc}}, \lambda y. \Phi(\text{pred } \alpha_{\text{tc}} y)(f' y))(a, f))) \\ &= \mathbf{T}_{\mathbb{M}}(\widehat{\mathbf{U}}_{f^{\mathbb{M}}[\text{index } \alpha_{\text{tc}}, \lambda y. \Phi(\text{pred } \alpha_{\text{tc}} y)(f' y), a, f]}) \\ &= \mathbf{U}_{f^{\mathbb{M}}[\text{index } \alpha_{\text{tc}}, \lambda y. \Phi(\text{pred } \alpha_{\text{tc}} y)(f' y), a, f]}. \end{aligned}$$

In this proof, we abbreviate $f^{\mathbb{M}}[\text{index } \alpha_{\text{tc}}, \lambda y. \Phi(\text{pred } \alpha_{\text{tc}} y)(f' y), a, f]$ as $f^{\mathbb{M}}$. Recall that $f^{\mathbb{M}}$ is of the form

$$\lambda w. (\widehat{\mathbf{N}}_{1\mathbb{M}} \widehat{+}_{\mathbb{M}} a \widehat{+}_{\mathbb{M}} (\text{index } \alpha_{\text{tc}}) \widehat{+}_{\mathbb{M}} \widehat{\Sigma}_{\mathbb{M}}(\text{index } \alpha_{\text{tc}}, \lambda w'. \mathbf{p}_1((\lambda y. \Phi(\text{pred } \alpha_{\text{tc}} y)(f' y)) w' w)), h^{\mathbb{M}} w)$$

by Definition 2.3.

On the other hand, the following holds for $\text{res}_{f^{\mathbb{M}}}^1(b, g, \text{ij } u) : \mathcal{U}_{(a,f)}^{\alpha, t}$.

$$\begin{aligned} \widehat{\mathbf{T}}_{(a,f)}^{\alpha, t}(\text{res}_{f^{\mathbb{M}}}^1(b, g, \text{ij } u)) &= \mathbf{p}_2(f^{\mathbb{M}}(b^*, g^*))(\text{ij } u) \\ &= h^{\mathbb{M}}(b^*, g^*)(\text{ij } u) = \widehat{\mathcal{U}} \end{aligned}$$

Therefore, $\text{res}_{f^{\mathbb{M}}}^1(b, g, \text{ij } u)$ is a code of $\widehat{\mathcal{U}}$ in $\mathcal{U}_{(a,f)}^{\alpha, t}$. Put $\widetilde{\mathcal{U}} := \text{res}_{f^{\mathbb{M}}}^1(b, g, \text{ij } u)$. Then, by defining

$$\widetilde{\mathbf{T}} := \lambda z. \text{res}_{f^{\mathbb{M}}}^1(b, g, \text{j}(u, z)) : \mathbf{T}_{(a,f)}^{\alpha, t}(\widetilde{\mathcal{U}}) \rightarrow \mathcal{U}_{(a,f)}^{\alpha, t},$$

we have $\widetilde{\mathcal{V}} := \widehat{\mathbf{W}}_{f^{\mathbb{M}}}(\widetilde{\mathcal{U}}, \widetilde{\mathbf{T}}) : \mathcal{U}_{(a,f)}^{\alpha, t}$.

Next, we show that $\xi \in \mathbf{V}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u}$ holds. Let $\widehat{\mathbf{V}} := \mathbf{V}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u}$. It suffices to verify that for any $\delta : \mathbb{V}$, if $\delta \in \beta$ holds then $\delta \in \widehat{\mathbf{V}}$ holds, because we have $\xi \in \beta$.

We prove the claim by transfinite induction on δ . Assume that $\delta \in \beta$ holds, so we have $\delta \doteq \mathbf{h}_{(a, f)}^{\alpha, t}(\text{pred } x z)$ for some $z : \mathbf{T}_{(a, f)}^{\alpha, t}(b)$. Let $C' := g z$. For any $v : \mathbf{T}_{(a, f)}^{\alpha, t}(C')$, we have

$$\begin{aligned} \mathbf{h}_{(a, f)}^{\alpha, t}(\text{pred}(\text{pred } x z) v) &\in \sup(\widehat{\mathbf{T}}_{(a, f)}^{\alpha, t}(\text{index}(\text{pred } x z)), \lambda y. \mathbf{h}_{(a, f)}^{\alpha, t}(\text{pred}(\text{pred } x z) y)) \\ &= \mathbf{h}_{(a, f)}^{\alpha, t} \sup(\text{index}(\text{pred } x z), \text{pred}(\text{pred } x z)) \\ &\doteq \mathbf{h}_{(a, f)}^{\alpha, t}(\text{pred } x z) \\ &\doteq \delta. \end{aligned}$$

Since β is transitive, we have $\mathbf{h}_{(a, f)}^{\alpha, t}(\text{pred}(\text{pred } x z) v) \in \beta$, so $\mathbf{h}_{(a, f)}^{\alpha, t}(\text{pred}(\text{pred } x z) v) \in \widehat{\mathbf{V}}$ by IH of induction on δ . Thus,

$$\Pi_{(v : \mathbf{T}_{(a, f)}^{\alpha, t}(C'))} \Sigma_{(y : \mathcal{V}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u})} \mathbf{h}_{(a, f)}^{\alpha, t}(\text{pred}(\text{pred } x z) v) \doteq \mathbf{h}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u} y$$

holds, hence we have by the type-theoretic axiom of choice provable in **MLTT**

$$\Sigma_{(g' : \mathbf{T}_{(a, f)}^{\alpha, t}(C') \rightarrow \mathcal{V}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u})} \Pi_{(v : \mathbf{T}_{(a, f)}^{\alpha, t}(C'))} \mathbf{h}_{(a, f)}^{\alpha, t}(\text{pred}(\text{pred } x z) v) \doteq \mathbf{h}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u}(g' v). \quad (3.1)$$

On the other hand, by Lemma 3.19, there is a term C of type $\mathcal{U}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u}$ such that $\widehat{\mathbf{T}}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u}(C) =_{\mathbb{M}} g^* z$ holds. Since $g^* z = \widehat{\mathbf{T}}_{(a, f)}^{\alpha, t}(g z) = \widehat{\mathbf{T}}_{(a, f)}^{\alpha, t}(C')$ holds, we have a term p of type $\widehat{\mathbf{T}}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u}(C) =_{\mathbb{M}} \widehat{\mathbf{T}}_{(a, f)}^{\alpha, t}(g z)$. Therefore, we also have a term

$$\sup(C, \lambda y. g'(\text{transport}^{\mathbf{T}_{\mathbb{M}}(v)} p y)) : \mathcal{V}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u},$$

because for any $y : \widehat{\mathbf{T}}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u}(C)$, we have $g'(\text{transport}^{\mathbf{T}_{\mathbb{M}}(v)} p y) : \mathcal{V}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u}$. So

$$\mathbf{h}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u} \sup(C, \lambda y. g'(\text{transport}^{\mathbf{T}_{\mathbb{M}}(v)} p y)) \in \widehat{\mathbf{V}} = \mathbf{V}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u}$$

holds.

Then, by (3.1) and Lemma 3.18,

$$\begin{aligned} &\mathbf{h}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u} \sup(C, \lambda y. g'(\text{transport}^{\mathbf{T}_{\mathbb{M}}(v)} p y)) \\ &\doteq \sup(\widehat{\mathbf{T}}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u}(C), \lambda y. \mathbf{h}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u}(g'(\text{transport}^{\mathbf{T}_{\mathbb{M}}(v)} p y))) \\ &\doteq \sup(\widehat{\mathbf{T}}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u}(C), \lambda y. \mathbf{h}_{(a, f)}^{\alpha, t}(\text{pred}(\text{pred } x z)(\text{transport}^{\mathbf{T}_{\mathbb{M}}(v)} p y))) \\ &\doteq \sup(\widehat{\mathbf{T}}_{(a, f)}^{\alpha, t}(C'), \lambda y. \mathbf{h}_{(a, f)}^{\alpha, t}(\text{pred}(\text{pred } x z) y)) \\ &\doteq \delta \end{aligned}$$

holds, so we have $\delta \in \widehat{\mathbf{V}}$.

Finally, we show that $\widehat{\mathbf{V}} \in \mathbf{V}_{(a, f)}^{\alpha, t}$ holds. Recall that $\widetilde{\mathcal{V}}$ is of type $\mathcal{U}_{(a, f)}^{\alpha, t}$, and put

$$\mathcal{V} := \mathbf{T}_{(a, f)}^{\alpha, t}(\widetilde{\mathcal{V}}) = \mathbf{W}_{(v : \widehat{\mathcal{U}})} \mathbf{T}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u}(v).$$

By recursion on $\mathcal{V}$, we define $l : \mathcal{V} \rightarrow \mathcal{V}_{(a, f)}^{\alpha, t}$ as

$$l \sup(d, q) := \sup(\widetilde{\mathbf{T}} d, \lambda y. l(q y)),$$

and put $\mathbf{V}' := \sup(\tilde{\mathcal{V}}, l) : \mathcal{V}_{(a,f)}^{\alpha,t}$.

We then define $\beta' := \mathbf{h}_{(a,f)}^{\alpha,t} \mathbf{V}' : \mathbb{V}$, so $\beta' \in \mathbf{V}_{(a,f)}^{\alpha,t}$ holds. It suffices to show that we have $\beta' \doteq \hat{\mathbf{V}}$.

We first show that

$$\Pi_{(v:\mathcal{V})} \mathbf{h}_{(a,f)}^{\alpha,t}(l v) \doteq \mathbf{h}_{(b^*,g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u} v \quad (3.2)$$

holds by induction on $v = \sup(c, q)$. It follows from the induction hypothesis that we have

$$\Pi_{(y:\mathbf{T}_{(b^*,g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u} c)} \mathbf{h}_{(a,f)}^{\alpha,t}(l(q y)) \doteq \mathbf{h}_{(b^*,g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u}(q y).$$

So,

$$\begin{aligned} \mathbf{h}_{(a,f)}^{\alpha,t}(l v) &= \mathbf{h}_{(a,f)}^{\alpha,t} \sup(\tilde{\mathbf{T}} c, \lambda y. l(q y)) \\ &= \sup(\hat{\mathbf{T}}_{(a,f)}^{\alpha,t}(\tilde{\mathbf{T}} c), \lambda y. \mathbf{h}_{(a,f)}^{\alpha,t}(l(q y))) \\ &\doteq \sup(\hat{\mathbf{T}}_{(a,f)}^{\alpha,t}(\tilde{\mathbf{T}} c), \lambda y. \mathbf{h}_{(b^*,g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u}(q y)) \\ &= \mathbf{h}_{(b^*,g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u} v \end{aligned}$$

holds. The last equation here holds because

$$\begin{aligned} \hat{\mathbf{T}}_{(b^*,g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u}(c) &= \mathbf{p}_2(\Phi_{(b^*,g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u} c) \\ &= h^{\mathbb{M}}(b^*, g^*)(j(u, c)) \text{ with } u : \overline{\alpha_{\text{tc}}} \\ &= \mathbf{p}_2(f^{\mathbb{M}}(b^*, g^*)) j(u, c) \\ &= \hat{\mathbf{T}}_{(a,f)}^{\alpha,t}(\text{res}_{f^{\mathbb{M}}}^1(b, g, j(u, c))) \\ &= \hat{\mathbf{T}}_{(a,f)}^{\alpha,t}(\tilde{\mathbf{T}} c). \end{aligned}$$

Therefore, we have (3.2).

Since $\beta' = \sup(\hat{\mathbf{T}}_{(a,f)}^{\alpha,t}(\tilde{\mathcal{V}}), \lambda y. \mathbf{h}_{(a,f)}^{\alpha,t}(l y))$ holds, if we have $\eta \in \beta'$ then $\eta \doteq \mathbf{h}_{(a,f)}^{\alpha,t}(l v)$ holds for some $v : \mathcal{V}$. We thus have $\eta \doteq \mathbf{h}_{(a,f)}^{\alpha,t}(l v) \doteq \mathbf{h}_{(b^*,g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u} v \in \hat{\mathbf{V}}$ by (3.2).

Conversely, if $\eta \in \hat{\mathbf{V}}$ holds then $\eta \doteq \mathbf{h}_{(b^*,g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u} v$ holds for some $v : \mathcal{V}$. So, we have

$$\begin{aligned} \eta &\doteq \mathbf{h}_{(b^*,g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u} v \\ &\doteq \mathbf{h}_{(a,f)}^{\alpha,t}(l v) \text{ by (3.2)} \\ &\in \beta'. \end{aligned}$$

It follows that $\beta' \doteq \hat{\mathbf{V}}$. □

We now prove the lemma saying that for any $\alpha : \mathbb{V}$ such that $\text{Acc } \alpha$ holds with $a : \mathbb{M}$ and $f : \mathbf{T}_{\mathbb{M}}(a) \rightarrow \mathbb{M}$, $\mathbf{V}_{(a,f)}^{\alpha,t}$ is α -set-inaccessible.

Lemma 3.21 (Link). *For all $\alpha : \mathbb{V}$, $t : \text{Acc } \alpha$, $a : \mathbb{M}$ and $f : \mathbf{T}_{\mathbb{M}}(a) \rightarrow \mathbb{M}$, $\mathbf{V}_{(a,f)}^{\alpha,t}$ is α -set-inaccessible.*

Proof. Induction on α followed by subinduction on t . Let $\beta \in \alpha$ be the case. We then have $\beta \in \alpha_{\text{tc}}$, so $\beta \doteq \text{pred } \alpha_{\text{tc}} u$ for some $u : \overline{\alpha_{\text{tc}}}$.

If $\xi \in \mathbf{V}_{(a,f)}^{\alpha, \text{prog } f'}$ holds then Lemma 3.20 implies that there are terms $b : \mathcal{U}_{(a,f)}^{\alpha, \text{prog } f'}$ and $g : \mathbf{T}_{(a,f)}^{\alpha, \text{prog } f'}(b) \rightarrow \mathcal{U}_{(a,f)}^{\alpha, \text{prog } f'}$ with

$$\xi \in \mathbf{V}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u} \in \mathbf{V}_{(a,f)}^{\alpha, \text{prog } f'},$$

where $b^* := \widehat{\mathbf{T}}_{(a,f)}^{\alpha, \text{prog } f'}(b)$ and $g^* := \lambda x. \widehat{\mathbf{T}}_{(a,f)}^{\alpha, \text{prog } f'}(g x)$.

By IH of main induction, $\mathbf{V}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u}$ is $(\text{pred } \alpha_{\text{tc}} u)$ -set-inaccessible. Since α -set-inaccessibility is the translation of a corresponding set-theoretic formula for any $\alpha : \mathbb{V}$, Lemma 3.11 implies that if $\delta \doteq \eta$ holds and γ is δ -set-inaccessible then γ is also η -set-inaccessible. Therefore, $\mathbf{V}_{(b^*, g^*)}^{\text{pred } \alpha_{\text{tc}} u, f' u}$ is also β -set-inaccessible, and it follows that β -set-inaccessibles are unbounded in $\mathbf{V}_{(a,f)}^{\alpha, \text{prog } f'}$. By Lemmas 3.5 and 3.17.(5), $\mathbf{V}_{(a,f)}^{\alpha, \text{prog } f'}$ is α -set-inaccessible. $\square$

Below we use the following notations:

$$\begin{aligned} \widehat{\mathcal{U}}_{\beta}^{\alpha, t} &:= \mathbf{p}_1(\Phi_{(\text{index } \beta, \lambda x. \text{index } (\text{pred } \beta x))}^{\alpha, t}) & \mathcal{U}_{\beta}^{\alpha, t} &:= \mathbf{T}_{\mathbb{M}} \widehat{\mathcal{U}}_{\beta}^{\alpha, t} \\ \widehat{\mathbf{T}}_{\beta}^{\alpha, t} &:= \mathbf{p}_2(\Phi_{(\text{index } \beta, \lambda x. \text{index } (\text{pred } \beta x))}^{\alpha, t}) & \mathbf{T}_{\beta}^{\alpha, t} &:= \lambda x. \mathbf{T}_{\mathbb{M}}(\widehat{\mathbf{T}}_{\beta}^{\alpha, t} x) \\ \widehat{\mathcal{V}}_{\beta}^{\alpha, t} &:= \widehat{\mathbf{W}}_{\mathbb{M}}(\widehat{\mathcal{U}}_{\beta}^{\alpha, t}, \widehat{\mathbf{T}}_{\beta}^{\alpha, t}) & \mathcal{V}_{\beta}^{\alpha, t} &:= \mathbf{T}_{\mathbb{M}} \widehat{\mathcal{V}}_{\beta}^{\alpha, t} \end{aligned}$$

We also define $\mathbf{h}_{\beta}^{\alpha, t}$ and $\mathbf{V}_{\beta}^{\alpha, t}$ as

$$\mathbf{h}_{\beta}^{\alpha, t} \text{sup}(b, g) := \text{sup}(\widehat{\mathbf{T}}_{\beta}^{\alpha, t}(b), \lambda x. \mathbf{h}_{\beta}^{\alpha, t}(g x)), \quad \mathbf{V}_{\beta}^{\alpha, t} := \text{sup}(\widehat{\mathcal{V}}_{\beta}^{\alpha, t}, \mathbf{h}_{\beta}^{\alpha, t}),$$

respectively.

Lemma 3.22 (Link). *For any $\beta : \mathbb{V}$, if β is transitive, then for any $\alpha : \mathbb{V}$ and any $t : \text{Acc } \alpha$, we have $\Pi_{(\gamma : \mathbb{V})}(\gamma \in \beta \rightarrow \gamma \in \mathbf{V}_{\beta}^{\alpha, t})$.*

Proof. We prove $\Pi_{(\gamma : \mathbb{V})}(\gamma \in \beta \rightarrow \gamma \in \mathbf{V}_{\beta}^{\alpha, t})$ by induction on γ . Suppose that $\gamma \in \beta$ holds. Then, $\gamma \doteq \text{pred } \beta x$ holds for some $x : \beta$. Put $\xi := \text{pred } \beta x$. If $u : \bar{\xi}$ holds, then we have $\text{pred } \xi u \in \gamma \in \beta$. Since β is transitive, we also have $\text{pred } \xi u \in \beta$ and so $\text{pred } \xi u \in \mathbf{V}_{\beta}^{\alpha, t}$ holds by IH. We thus have

$$\Pi_{(u : \bar{\xi})} \Sigma_{(z : \mathcal{V}_{\beta}^{\alpha, t})} \text{pred } \xi u \doteq \mathbf{h}_{\beta}^{\alpha, t} z.$$

By the type-theoretic axiom of choice in **MLTT**,

$$\Sigma_{(f : \bar{\xi} \rightarrow \mathcal{V}_{\beta}^{\alpha, t})} \Pi_{(u : \bar{\xi})} \text{pred } \xi u \doteq \mathbf{h}_{\beta}^{\alpha, t}(f u)$$

holds.

On the other hand, by Lemma 3.19, there is a term $c : \mathcal{U}_{\beta}^{\alpha, t}$ such that

$$p : \widehat{\mathbf{T}}_{\beta}^{\alpha, t}(c) =_{\mathbb{M}} (\lambda z. \text{index } (\text{pred } \beta z))x$$

holds. Then, it follows from Lemma 3.18 that

$$\text{sup}(\widehat{\mathbf{T}}_{\beta}^{\alpha, t}(c), \lambda x. \mathbf{h}_{\beta}^{\alpha, t}(f(\text{transport}^{\mathbf{T}_{\mathbb{M}}(v)} p x))) \doteq \text{sup}(\text{index } \xi, \lambda u. \mathbf{h}_{\beta}^{\alpha, t}(f u))$$

holds. Note that we have

$$\text{sup}(c, \lambda x. f(\text{transport}^{\mathbf{T}_{\mathbb{M}}(v)} p x)) : \mathcal{V}_{\beta}^{\alpha, t}, \quad \mathbf{h}_{\beta}^{\alpha, t} \text{sup}(c, \lambda x. f(\text{transport}^{\mathbf{T}_{\mathbb{M}}(v)} p x)) \in \mathbf{V}_{\beta}^{\alpha, t}.$$

Moreover,

$$\begin{aligned}
\mathbf{h}_\beta^{\alpha,t} \text{sup}(c, \lambda x. f (\text{transport}^{\text{T}_\mathbb{M}(v)} p x)) &= \text{sup}(\widehat{\text{T}}_\beta^{\alpha,t} c, \lambda x. \mathbf{h}_\beta^{\alpha,t} (f (\text{transport}^{\text{T}_\mathbb{M}(v)} p x))) \\
&\doteq \text{sup}(\text{index } \xi, \lambda u. \mathbf{h}_\beta^{\alpha,t} (f u)) \\
&\doteq \text{sup}(\text{index } \xi, \text{pred } \xi) \\
&\doteq \xi
\end{aligned}$$

holds, hence we have $\gamma \doteq \xi \in \mathbf{V}_\beta^{\alpha,t}$. □

Lemma 3.23 (Link). *The following holds:*

- (1) *We have a term $\text{inv} : \Pi_{(\alpha:\mathbb{V})} (\text{Acc } \alpha \rightarrow \Pi_{(x:\overline{\alpha_{\text{tc}}})} \text{Acc } (\text{pred } \alpha_{\text{tc}} x))$.*
- (2) *We have a function $\text{acc} : \Pi_{(\alpha:\mathbb{V})} \text{Acc } \alpha$.*

Proof. (1) By induction on $t : \text{Acc } \alpha$. Let t be $\text{prog } f$, then we have $f x : \text{Acc } (\text{pred } \alpha_{\text{tc}} x)$.

(2) By induction on $\alpha : \mathbb{V}$. Let α be $\text{sup}(a, f)$, and recall that $\overline{(\text{sup}(a, f))_{\text{tc}}}$ is definitionally equal to $\text{T}_\mathbb{M}(a) + \Sigma_{(z:\text{T}_\mathbb{M}(a))} \overline{(f z)_{\text{tc}}}$. Define $g : \Pi_{(x:\overline{(\text{sup}(a, f))_{\text{tc}}})} \text{Acc } (\text{pred } (\text{sup}(a, f))_{\text{tc}} x)$ by

$$\begin{aligned}
g(\text{i } z) &= \text{acc } (f z), \\
g(\text{j } w) &= \text{inv } (f (\text{p}_1 w)) (\text{acc } (f (\text{p}_1 w))) (\text{p}_2 w).
\end{aligned}$$

The function g is well-defined because we have

$$\text{pred } (\text{sup}(a, f))_{\text{tc}} (\text{i } z) = f z, \quad \text{pred } (\text{sup}(a, f))_{\text{tc}} (\text{j } w) = \text{pred } (f (\text{p}_1 w))_{\text{tc}} (\text{p}_2 w).$$

We then put $\text{acc } (\text{sup}(a, f)) := \text{prog } g$. □

We have an interpretation of **Pi-Numbers** in $\mathbf{MLM}_{\text{acc}}$ by the theorem below:

Theorem 3.24 (Link). *For any $\gamma, \alpha : \mathbb{V}$, there is a term $\beta : \mathbb{V}$ such that $\gamma \in \beta$ holds and β is α -set-inaccessible.*

Proof. Let $\gamma, \alpha : \mathbb{V}$ be given. It is straightforward to define a term γ' of $\mathbb{V}$ with $\gamma \in \gamma'$, since all axioms of **CZF** are interpretable in **MLTT**. By Lemmas 3.22 and 3.23.(2), we have

$$\Pi_{(\delta:\mathbb{V})} (\delta \in \gamma'_{\text{tc}} \rightarrow \delta \in \mathbf{V}_{\gamma'_{\text{tc}}}^{\alpha, \text{acc } \alpha}).$$

Therefore, $\gamma \in \mathbf{V}_{\gamma'_{\text{tc}}}^{\alpha, \text{acc } \alpha}$ holds, and $\mathbf{V}_{\gamma'_{\text{tc}}}^{\alpha, \text{acc } \alpha}$ is α -set-inaccessible by Lemma 3.21. □

The statements whose proofs are not formalised yet in [Tak25] are as follows:

- (1) $\mathbf{V}_{(a,f)}^{\alpha,t}$ satisfies **Regular Extension Axiom** (cf. [RGP98]).
- (2) For any $\alpha, \beta : \mathbb{V}$, if $\alpha \doteq \beta$ holds and α is set-inaccessible then β is set-inaccessible.
- (3) For any $\alpha, \beta, \gamma : \mathbb{V}$, if $\alpha \doteq \beta$ holds and γ is α -set-inaccessible then γ is β -set-inaccessible.

The statement 1 is postulated in `RegularExtensionAxiom.agda` of [Tak25], while the other statements are postulated in `Inaccessibles/Basics.agda`. The statement 1 is used to prove Lemma 3.17.(5), and we referred to [RGP98, Corollary 4.8] for its proof. Postulating the statements 2 and 3 is harmless, since we have Lemma 3.11 for the **MLTT**-translation of the set-theoretic terms “set-inaccessible” and “ α -set-inaccessible”.

4. CONCLUDING REMARK

In this paper, we showed that the system $\mathbf{CZF}_\pi$ of Aczel’s constructive set theory with inaccessible sets is interpretable in a variant of $\mathbf{MLTT}$ equipped with a Mahlo universe and accessibility predicate Acc . Our type-theoretic counterparts of inaccessible sets were constructed by the transfinite iteration of a higher-order universe operator whose definition used the reflection property of the Mahlo universe. The accessibility predicate Acc provided a recursion principle to iterate this operator. Moreover, we formalised the main part of this interpretation in Agda. For this purpose, the support of indexed induction-recursion in Agda is crucial: the Mahlo universe was simulated by the external Mahlo universe defined with induction-recursion, and the predicate Acc was defined by indexed inductive definition.

A future research direction is to investigate the extent to which the transfinite hierarchy of inaccessible sets can be extended in $\mathbf{MLTT}$. In set-theoretic terms, the diagonalisation of this hierarchy gives the class $\{a \mid a \text{ is } a\text{-set-inaccessible}\}$ (cf. [RA74]). Members of this class can be considered as “fixed points” of an increasing sequence from inaccessible sets. It would be interesting to examine whether $\mathbf{MLM}$ can construct an element of $\mathbb{V}$ corresponding to a member of this class and, if possible, whether such elements of $\mathbb{V}$ are unbounded in $\mathbb{V}$.

Another direction is to revisit our construction of inaccessible sets in $\mathbf{MLM}$ from the viewpoint of homotopy type theory (HoTT). In [Uni13], Aczel’s interpretation of $\mathbf{CZF}$ in $\mathbf{MLTT}$ was refined: the type of Aczel’s iterative sets was defined as a higher inductive type $\mathbb{V}_h$. In addition to its ordinary introduction rule (*i.e.*, its point constructor rule), which is the same as that of $\mathbb{V}$, the higher inductive type $\mathbb{V}_h$ also has path constructor rules. Although it is unknown whether the axioms **Strong Collection** and **Subset Collection** of $\mathbf{CZF}$ hold in this refined interpretation, several interesting results on this interpretation have already been obtained (see, *e.g.*, [dJKFX23]). Later, a general framework for interpreting constructive set theory into HoTT was proposed in [Gal21]. Importantly, all axioms of $\mathbf{CZF}$ hold in some interpretations of this framework; therefore, it would provide a HoTT framework to revisit our interpretation of $\mathbf{CZF}_\pi$ in $\mathbf{MLM}_{\text{acc}}$.

ACKNOWLEDGEMENT

The author is grateful to the anonymous reviewers for their comments and suggestions, which improved the contents and presentation of the present paper. The work in this paper is supported by JSPS KAKENHI Grant Number JP21K12822.

REFERENCES

- [Acz78] Peter Aczel. The type theoretic interpretation of constructive set theory. In Angus Macintyre, Leszek Pacholski, and Jeff Paris, editors, *Logic Colloquium ’77*, volume 96 of *Studies in Logic and the Foundations of Mathematics*, pages 55–66. Elsevier, 1978. doi:10.1016/S0049-237X(08)71989-X.
- [Acz82] Peter 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.
- [Acz86] Peter Aczel. The type theoretic interpretation of constructive set theory: Inductive definitions. In R. B. Marcus, G. J. Dorn, and G. J. W. Dorn, editors, *Logic, Methodology, and Philosophy of Science VII*, pages 17–49. North-Holland, 1986.
- [Agd25] AgdaTeam. The Agda Wiki, 2025. <https://wiki.portal.chalmers.se/agda/pmwiki.php>.
- [AR01] Peter Aczel and Michael Rathjen. Notes on constructive set theory. Technical Report 40, Institut Mittag-Leffler, Djursholm, 2001.

- [dJKFX23] Tom de Jong, Nicolai Kraus, Fredrik Nordvall Forsberg, and Chuangjie Xu. Set-theoretic and type-theoretic ordinals coincide. In *LICS*, pages 1–13, 2023. doi:10.1109/LICS56636.2023.10175762.
- [DP20] Peter Dybjer and Erik Palmgren. Intuitionistic Type Theory. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2020 edition, 2020. URL: <https://plato.stanford.edu/archives/sum2020/entries/type-theory-intuitionistic/>.
- [Dra74] Frank R. Drake. *Set Theory*, volume 76 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 1974. doi:10.1016/S0049-237X(08)71774-9.
- [DS03] Peter Dybjer and Anton Setzer. Induction-recursion and initial algebras. *Ann. Pure Appl. Log.*, 124(1-3):1–47, 2003. doi:10.1016/S0168-0072(02)00096-9.
- [DS06] Peter Dybjer and Anton Setzer. Indexed induction-recursion. *J. Log. Algebraic Methods Program.*, 66(1):1–49, 2006. doi:10.1016/J.JLAP.2005.07.001.
- [Dyb00] Peter Dybjer. A general formulation of simultaneous inductive-recursive definitions in type theory. *J. Symb. Log.*, 65(2):525–549, 2000. doi:10.2307/2586554.
- [Gal21] Cesare Gallozzi. Homotopy type-theoretic interpretations of constructive set theories. *Math. Struct. Comput. Sci.*, 31(1):112–143, 2021. doi:10.1017/S0960129519000148.
- [GR94] Edward R. Griffor and Michael Rathjen. The strength of some Martin-Löf type theories. *Arch. Math. Log.*, 33(5):347–385, 1994. doi:10.1007/BF01278464.
- [Hof97] Martin Hofmann. Syntax and semantics of dependent types. In Andrew M. Pitts and P. Dybjer, editors, *Semantics and Logics of Computation*, Publications of the Newton Institute, pages 79–130. Cambridge University Press, Cambridge, 1997. doi:10.1017/CB09780511526619.004.
- [Hug21] Jasper Hugunin. Why not W? In Ugo de’Liguoro, Stefano Berardi, and Thorsten Altenkirch, editors, *26th International Conference on Types for Proofs and Programs, TYPES 2020, March 2-5, 2020, University of Turin, Italy*, volume 188 of *LIPICs*, pages 8:1–8:9. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICs.TYPES.2020.8.
- [Mah11] Paul Mahlo. Über lineare transfinite Mengen. *Berichte über die Verhandlungen der Königlich Sächsischen Gesellschaft der Wissenschaften zu Leipzig. Mathematisch-Physische Klasse*, 63:187–225, 1911.
- [ML84] Per Martin-Löf. *Intuitionistic type theory: Notes by Giovanni Sambin of a series of lectures given in Padua, June 1980*. Bibliopolis, 1984.
- [NPS90] Bengt Nordström, Kent Petersson, and Jan M. Smith. *Programming in Martin-Löf’s type theory. An introduction*. Oxford Univ. Press, 1990.
- [Pal98] Erik Palmgren. On universes in type theory. In Giovanni Sambin and Jan M. Smith, editors, *Twenty Five Years of Constructive Type Theory*, Oxford Logic Guides, pages 191–204. Oxford University Press, 1998.
- [RA74] Wayne Richter and Peter Aczel. Inductive definitions and reflecting properties of admissible ordinals. In J.E. Fenstad and P.G. Hinman, editors, *Generalized Recursion Theory*, volume 79 of *Studies in Logic and the Foundations of Mathematics*, pages 301–381. Elsevier, 1974. doi:10.1016/S0049-237X(08)70592-5.
- [Rat00] Michael Rathjen. The superjump in Martin-Löf type theory. In Samuel R. Buss, Petr Hájek, and Pavel Pudlák, editors, *Logic Colloquium ’98*, number 13 in *Lecture Notes in Logic*, pages 363–386. Cambridge University Press, Cambridge, 2000. doi:10.1017/9781316756140.027.
- [Rat03] Michael Rathjen. Realizing Mahlo set theory in type theory. *Arch. Math. Log.*, 42(1):89–101, 2003. doi:10.1007/s00153-002-0159-6.
- [RGP98] Michael Rathjen, Edward R. Griffor, and Erik Palmgren. Inaccessibility in constructive set theory and type theory. *Ann. Pure Appl. Log.*, 94(1-3):181–200, 1998. doi:10.1016/S0168-0072(97)00072-9.
- [Set93] Anton Setzer. *Proof theoretical strength of Martin-Löf Type Theory with W-type and one universe*. PhD thesis, Universität München, 1993.
- [Set98] Anton Setzer. Well-ordering proofs for Martin-Löf type theory. *Ann. Pure Appl. Logic*, 92(2):113–159, 1998. doi:10.1016/S0168-0072(97)00078-X.
- [Set00] Anton Setzer. Extending Martin-Löf type theory by one Mahlo-universe. *Arch. Math. Log.*, 39(3):155–181, 2000. doi:10.1007/s0015300050140.

- [Set08a] Anton Setzer. Proof theory and Martin-Löf Type Theory. In M. van Atten, P. Boldini, M. Bourdeau, and G. Heinzmann, editors, *One Hundred Years of Intuitionism (1907–2007)*, pages 257–279. Birkhäuser, 2008. doi:10.1007/978-3-7643-8653-5_16.
- [Set08b] Anton Setzer. Universes in type theory part I – Inaccessibles and Mahlo. In A. Andretta, K. Kearnes, and D. Zambella, editors, *Logic Colloquium '04*, pages 123–156. Association of Symbolic Logic, Lecture Notes in Logic 29, Cambridge University Press, 2008. doi:10.1017/CB09780511721151.009.
- [Tak25] Yuta Takahashi. CZF-in-Mahlo, 2025. <https://github.com/takahashi-yt/czf-in-mahlo>. doi:10.5281/zenodo.16901729.
- [Uni13] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. <https://homotopytypetheory.org/book>, Institute for Advanced Study, 2013.

APPENDIX A. INFERENCE RULES OF MLM

In this appendix, we define the inference rules of **MLM** in detail to make the present paper self-contained as far as possible. We provided the Agda code corresponding to the rules for the Mahlo universe $\mathbb{M}$ in `ExternalMahlo.agda` of [Tak25].

Definition A.1 (Inference Rules and Judgements in **MLM**). Judgements in **MLM** are those which are derivable by the following inference rules of **MLM**:

Rules for Contexts: let x be fresh for Γ , and y be fresh for Δ .

$$\frac{}{\vdash \emptyset \text{ ctxt}} \quad \frac{\Gamma \vdash A \text{ type}}{\vdash \Gamma, x : A \text{ ctxt}}$$

Rule for Assumptions:

$$\frac{\vdash \Gamma, x : A, \Delta \text{ ctxt}}{\Gamma, x : A, \Delta \vdash x : A}$$

Rules for Substitutions:

$$\frac{\Gamma \vdash a : A \quad \Gamma, x : A, \Delta \vdash \theta}{\Gamma, \Delta[a/x] \vdash \theta[a/x]} \quad \frac{\Gamma \vdash a = b : A \quad \Gamma, x : A, \Delta \vdash B \text{ type}}{\Gamma, \Delta[a/x] \vdash B[a/x] = B[b/x]}$$

$$\frac{\Gamma \vdash a = b : A \quad \Gamma, x : A, \Delta \vdash c : B}{\Gamma, \Delta[a/x] \vdash c[a/x] = c[b/x] : B[a/x]}$$

Hereafter, we list the premises of an inference rule in the vertical way.

Rules for Definitional Equality:

$$\frac{\Gamma \vdash a : A}{\Gamma \vdash a = a : A} \quad \frac{\Gamma \vdash a = b : A}{\Gamma \vdash b = a : A} \quad \frac{\Gamma \vdash a = b : A \quad \Gamma \vdash b = c : A}{\Gamma \vdash a = c : A}$$

$$\frac{\Gamma \vdash A \text{ type}}{\Gamma \vdash A = A} \quad \frac{\Gamma \vdash A = B}{\Gamma \vdash B = A} \quad \frac{\Gamma \vdash A = B \quad \Gamma \vdash B = C}{\Gamma \vdash A = C} \quad \frac{\Gamma \vdash a : A \quad \Gamma \vdash A = B}{\Gamma \vdash a : B}$$

Rules for Π -Types: we have the formation rule Πfm , the introduction rule Πin , the elimination rule Πel and the β/η equality rules $\Pi\text{eq}\beta/\Pi\text{eq}\eta$ for Π -types.

$$\frac{\Gamma \vdash A \text{ type} \quad \Gamma, x : A \vdash B \text{ type}}{\Gamma \vdash \Pi_{(x:A)} B \text{ type}} \Pi\text{fm} \quad \frac{\Gamma, x : A \vdash b : B}{\Gamma \vdash \lambda x. b : \Pi_{(x:A)} B} \Pi\text{in}$$

$$\begin{array}{c}
\Gamma, x : A \vdash B \text{ type} \\
\Gamma \vdash b : \Pi_{(x:A)} B \\
\hline
\Gamma \vdash a : A \\
\hline
\Gamma \vdash b a : B[a/x] \quad \Pi\text{el}
\end{array}
\quad
\begin{array}{c}
\Gamma, x : A \vdash b : B \\
\Gamma \vdash a : A \\
\hline
\Gamma \vdash (\lambda x. b) a = b[a/x] : B[a/x] \quad \Pi\text{eq}\beta
\end{array}$$

$$\begin{array}{c}
\Gamma, x : A \vdash B \text{ type} \\
\Gamma \vdash b : \Pi_{(x:A)} B \\
\hline
\Gamma \vdash \lambda y. b y = b : \Pi_{(x:A)} B \quad \Pi\text{eq}\eta \quad \text{with } y \text{ not free in } b
\end{array}$$

Moreover, we have a congruence rule for each of Πfm , Πin , Πel as follows.

$$\begin{array}{c}
\Gamma \vdash A_1 = A_2 \\
\Gamma, x : A_1 \vdash B_1 = B_2 \\
\hline
\Gamma \vdash \Pi_{(x:A_1)} B_1 = \Pi_{(x:A_2)} B_2 \quad \Pi\text{fm}_c
\end{array}
\quad
\begin{array}{c}
\Gamma, x : A \vdash b_1 = b_2 : B \\
\hline
\Gamma \vdash \lambda x. b_1 = \lambda x. b_2 : \Pi_{(x:A)} B \quad \Pi\text{in}_c
\end{array}$$

$$\begin{array}{c}
\Gamma, x : A \vdash B \text{ type} \\
\Gamma \vdash b_1 = b_2 : \Pi_{(x:A)} B \\
\Gamma \vdash a_1 = a_2 : A \\
\hline
\Gamma \vdash b_1 a_1 = b_2 a_2 : B[a_1/x] \quad \Pi\text{el}_c
\end{array}$$

As in the case of Π -types, for any $*$ $\in \{\Sigma, +, N_n, N, W, \text{Id}\}$, we use the labels $*\text{fm}$, $*\text{in}$, $*\text{el}$ and $*\text{eq}$ possibly with suffixes to denote $*$ -formation rules, $*$ -introduction rules, $*$ -elimination rules and $*$ -equality rules, respectively. Hereafter, we omit to write down congruence rules: we assume that congruence rules are formulated for the $*$ -formation, $*$ -introduction and $*$ -elimination rules.

Rules for Σ -Types: we have the following rules.

$$\begin{array}{c}
\Gamma \vdash A \text{ type} \\
\Gamma, x : A \vdash B \text{ type} \\
\hline
\Gamma \vdash \Sigma_{(x:A)} B \text{ type} \quad \Sigma\text{fm}
\end{array}
\quad
\begin{array}{c}
\Gamma, x : A \vdash B \text{ type} \\
\Gamma \vdash a : A \\
\Gamma \vdash b : B[a/x] \\
\hline
\Gamma \vdash (a, b) : \Sigma_{(x:A)} B \quad \Sigma\text{in}
\end{array}
\quad
\begin{array}{c}
\Gamma, z : \Sigma_{(x:A)} B \vdash C \text{ type} \\
\Gamma \vdash c : \Pi_{(x:A)} \Pi_{(y:B)} C[(x, y)/z] \\
\Gamma \vdash b : \Sigma_{(x:A)} B \\
\hline
\Gamma \vdash E_\Sigma(b, c) : C[b/z] \quad \Sigma\text{el}
\end{array}$$

$$\begin{array}{c}
\Gamma, x : A \vdash B \text{ type} \\
\Gamma \vdash a : A \\
\Gamma \vdash b : B[a/x] \\
\Gamma, z : \Sigma_{(x:A)} B \vdash C \text{ type} \\
\Gamma \vdash c : \Pi_{(x:A)} \Pi_{(y:B)} C[(x, y)/z] \\
\hline
\Gamma \vdash E_\Sigma((a, b), c) = c a b : C[(a, b)/z] \quad \Sigma\text{eq}
\end{array}$$

It is straightforward to define the left projection $\mathbf{p}_1$ and the right projection $\mathbf{p}_2$ satisfying the following:

$$\begin{array}{c}
\Gamma, x : A \vdash B \text{ type} \\
\Gamma \vdash c : \Sigma_{(x:A)} B \\
\hline
\Gamma \vdash \mathbf{p}_1 c : A
\end{array}
\quad
\begin{array}{c}
\Gamma, x : A \vdash B \text{ type} \\
\Gamma \vdash a : A \\
\Gamma \vdash b : B[a/x] \\
\hline
\Gamma \vdash \mathbf{p}_1(a, b) = a : A
\end{array}$$

$$\begin{array}{c}
\Gamma, x : A \vdash B \text{ type} \\
\Gamma \vdash c : \Sigma_{(x:A)} B \\
\hline
\Gamma \vdash \mathbf{p}_2 c : B[\mathbf{p}_1(c)/x]
\end{array}
\quad
\begin{array}{c}
\Gamma, x : A \vdash B \text{ type} \\
\Gamma \vdash a : A \\
\Gamma \vdash b : B[a/x] \\
\hline
\Gamma \vdash \mathbf{p}_2(a, b) = b : B[\mathbf{p}_1(a, b)/x]
\end{array}$$

Rules for $+$ -Types: a sum type $A + B$ corresponds to a disjoint sum of A and B .

$$\begin{array}{c}
\Gamma \vdash A \text{ type} \\
\Gamma \vdash B \text{ type} \\
\hline
\Gamma \vdash A + B \text{ type} \quad +\text{fm}
\end{array}
\quad
\begin{array}{c}
\Gamma \vdash a : A \\
\Gamma \vdash B \text{ type} \\
\hline
\Gamma \vdash \mathbf{i}(a) : A + B \quad +\text{in}_i
\end{array}
\quad
\begin{array}{c}
\Gamma \vdash A \text{ type} \\
\Gamma \vdash b : B \\
\hline
\Gamma \vdash \mathbf{j}(b) : A + B \quad +\text{in}_j
\end{array}$$

$$\frac{\begin{array}{c} \Gamma, x : A + B \vdash C \text{ type} \\ \Gamma \vdash c_1 : \Pi_{(y:A)} C[i(y)/x] \\ \Gamma \vdash c_2 : \Pi_{(z:B)} C[j(z)/x] \\ \Gamma \vdash d : A + B \end{array}}{\Gamma \vdash E_+(d, c_1, c_2) : C[d/x]} +\text{el}$$

$$\frac{\begin{array}{c} \Gamma, x : A + B \vdash C \text{ type} \\ \Gamma \vdash c_1 : \Pi_{(y:A)} C[i(y)/x] \\ \Gamma \vdash c_2 : \Pi_{(z:B)} C[j(z)/x] \\ \Gamma \vdash a : A \end{array}}{\Gamma \vdash E_+(i(a), c_1, c_2) = c_1 a : C[i(a)/x]} +\text{eq}_i \quad \frac{\begin{array}{c} \Gamma, x : A + B \vdash C \text{ type} \\ \Gamma \vdash c_1 : \Pi_{(y:A)} C[i(y)/x] \\ \Gamma \vdash c_2 : \Pi_{(z:B)} C[j(z)/x] \\ \Gamma \vdash b : B \end{array}}{\Gamma \vdash E_+(j(b), c_1, c_2) = c_2 b : C[j(b)/x]} +\text{eq}_j$$

Rules for N-Type: the N-type is the type of natural numbers. Its elimination rule is the induction principle on N.

$$\frac{\vdash \Gamma \text{ ctxt}}{\Gamma \vdash N \text{ type}} \text{Nfm} \quad \frac{\vdash \Gamma \text{ ctxt}}{\Gamma \vdash 0 : N} \text{Nin}_0 \quad \frac{\Gamma \vdash a : N}{\Gamma \vdash \text{suc}(a) : N} \text{Nin}_{\text{suc}}$$

$$\frac{\begin{array}{c} \Gamma, x : N \vdash C \text{ type} \\ \Gamma \vdash a : N \\ \Gamma \vdash b : C[0/x] \\ \Gamma \vdash c : \Pi_{(x:N)} (C \rightarrow C[\text{suc}(x)/x]) \end{array}}{\Gamma \vdash E_N(a, b, c) : C[a/x]} \text{Nel} \quad \frac{\begin{array}{c} \Gamma, x : N \vdash C \text{ type} \\ \Gamma \vdash b : C[0/x] \\ \Gamma \vdash c : \Pi_{(x:N)} (C \rightarrow C[\text{suc}(x)/x]) \end{array}}{\Gamma \vdash E_N(0, b, c) = b : C[0/x]} \text{Neq}_0$$

$$\frac{\begin{array}{c} \Gamma, x : N \vdash C \text{ type} \\ \Gamma \vdash a : N \\ \Gamma \vdash b : C[0/x] \\ \Gamma \vdash c : \Pi_{(x:N)} (C \rightarrow C[\text{suc}(x)/x]) \end{array}}{\Gamma \vdash E_N(\text{suc}(a), b, c) = c a (E_N(a, b, c)) : C[\text{suc}(a)/x]} \text{Neq}_{\text{suc}}$$

Rules for N_n-Type: the N_n-type is the type of finite set with n members.

$$\frac{\vdash \Gamma \text{ ctxt}}{\Gamma \vdash N_n \text{ type}} \text{N}_n\text{fm} \quad \frac{\vdash \Gamma \text{ ctxt}}{\Gamma \vdash m_n : N_n} \text{N}_n\text{in} \quad \text{for each numeral } m < n$$

$$\frac{\begin{array}{c} \Gamma, x : N_n \vdash C \text{ type} \\ \Gamma \vdash a : N_n \\ \Gamma \vdash b_0 : C[0_n/x] \\ \vdots \\ \Gamma \vdash b_{n-1} : C[(n-1)_n/x] \end{array}}{\Gamma \vdash E_n(a, b_0, \dots, b_{n-1}) : C[a/x]} \text{N}_n\text{el} \quad \frac{\begin{array}{c} \Gamma, x : N_n \vdash C \text{ type} \\ \Gamma \vdash b_0 : C[0_n/x] \\ \vdots \\ \Gamma \vdash b_{n-1} : C[(n-1)_n/x] \end{array}}{\Gamma \vdash E_n(m_n, b_0, \dots, b_{n-1}) = b_m : C[m_n/x]} \text{N}_n\text{eq}$$

Rules for W-Types: a W-type $W_{(x:A)}B$ corresponds to the type of well founded trees such that each of their branchings is indexed by $B[a/x]$ for some $a : A$. Its elimination rule is the induction principle on a well-founded tree.

$$\frac{\begin{array}{c} \Gamma \vdash A \text{ type} \\ \Gamma, x : A \vdash B \text{ type} \end{array}}{\Gamma \vdash W_{(x:A)}B \text{ type}} \text{Wfm} \quad \frac{\begin{array}{c} \Gamma, x : A \vdash B \text{ type} \\ \Gamma \vdash a : A \\ \Gamma \vdash b : B[a/x] \rightarrow W_{(x:A)}B \end{array}}{\Gamma \vdash \text{sup}(a, b) : W_{(x:A)}B} \text{Win}$$

$$\begin{array}{c}
\Gamma, x : A \vdash B \text{ type} \\
\Gamma, y' : W_{(x:A)}B \vdash C \text{ type} \\
\Gamma \vdash a : W_{(x:A)}B \\
\Gamma \vdash b : \Pi_{(y:A)}\Pi_{(z:B[y/x] \rightarrow W_{(x:A)}B)}(\Pi_{(v:B[y/x])}C[z \ v/y'] \rightarrow C[\text{sup}(y, z)/y']) \\
\hline
\Gamma \vdash E_W(a, b) : C[a/y'] \quad \text{Wel}
\end{array}$$

$$\begin{array}{c}
\Gamma, x : A \vdash B \text{ type} \\
\Gamma, y' : W_{(x:A)}B \vdash C \text{ type} \\
\Gamma \vdash a : A \\
\Gamma \vdash b : B[a/x] \rightarrow W_{(x:A)}B \\
\Gamma \vdash c : \Pi_{(y:A)}\Pi_{(z:B[y/x] \rightarrow W_{(x:A)}B)}(\Pi_{(v:B[y/x])}C[z \ v/y'] \rightarrow C[\text{sup}(y, z)/y']) \\
\hline
\Gamma \vdash E_W(\text{sup}(a, b), c) = c \ a \ b \ (\lambda v. E_W(\text{app}(b, v), c)) : C[\text{sup}(a, b)/y'] \quad \text{Weq}
\end{array}$$

Rules for Id-Types: an Id-type $\text{Id}(A, a_1, a_2)$ corresponds to an identity statement which says that a_1 and a_2 are equal elements of A .

$$\begin{array}{c}
\Gamma \vdash A \text{ type} \\
\Gamma \vdash a_1 : A \\
\Gamma \vdash a_2 : A \\
\hline
\Gamma \vdash \text{Id}(A, a_1, a_2) \text{ type} \quad \text{Idfm}
\end{array}
\quad
\begin{array}{c}
\Gamma \vdash a : A \\
\hline
\Gamma \vdash \text{refl}(a) : \text{Id}(A, a, a) \quad \text{Idin}
\end{array}$$

$$\begin{array}{c}
\Gamma, x : A, y : A, z : \text{Id}(A, x, y) \vdash C \text{ type} \\
\Gamma \vdash c : \Pi_{(v:A)}C[v/x, v/y, \text{refl}(v)/z] \\
\Gamma \vdash a : A \\
\Gamma \vdash b : A \\
\Gamma \vdash d : \text{Id}(A, a, b) \\
\hline
\Gamma \vdash E_{\text{Id}}(d, c) : C[a/x, b/y, d/z] \quad \text{Idel}
\end{array}$$

$$\begin{array}{c}
\Gamma, x : A, y : A, z : \text{Id}(A, x, y) \vdash C \text{ type} \\
\Gamma \vdash c : \Pi_{(v:A)}C[v/x, v/y, \text{refl}(v)/z] \\
\Gamma \vdash a : A \\
\hline
\Gamma \vdash E_{\text{Id}}(\text{refl}(a), c) = c \ a : C[a/x, a/y, \text{refl}(a)/z] \quad \text{Ideq}
\end{array}$$

Rules for Mahlo Universe ($\mathbb{M}, T_{\mathbb{M}}$): we have the rules $\mathbb{M}\text{fm}$, $\mathbb{M}\text{in}$ and $T_{\mathbb{M}}\text{fm}$ with the inference rules which define the decoding function $T_{\mathbb{M}}$ recursively. Congruence rules are defined for the $T_{\mathbb{M}}$ -formation rule $T_{\mathbb{M}}\text{fm}$ and all $\mathbb{M}$ -introduction rules $\mathbb{M}\text{in}$ except $\mathbb{M}\text{in}_N$ and $\mathbb{M}\text{in}_{N_n}$, though we omit to write down them.

$$\begin{array}{c}
\frac{\vdash \Gamma \text{ ctxt}}{\Gamma \vdash \mathbb{M} \text{ type}} \quad \mathbb{M}\text{fm} \quad \frac{\Gamma \vdash a : \mathbb{M}}{\Gamma \vdash T_{\mathbb{M}}(a) \text{ type}} \quad T_{\mathbb{M}}\text{fm}
\end{array}$$

$$\begin{array}{c}
\frac{\vdash \Gamma \text{ ctxt}}{\Gamma \vdash \widehat{N}_{\mathbb{M}} : \mathbb{M}} \quad \mathbb{M}\text{in}_N \quad \frac{\vdash \Gamma \text{ ctxt}}{\Gamma \vdash T_{\mathbb{M}}(\widehat{N}_{\mathbb{M}}) = N}
\end{array}$$

$$\begin{array}{c}
\frac{\vdash \Gamma \text{ ctxt}}{\Gamma \vdash \widehat{N}_{n\mathbb{M}} : \mathbb{M}} \quad \mathbb{M}\text{in}_{N_n} \quad \frac{\vdash \Gamma \text{ ctxt}}{\Gamma \vdash T_{\mathbb{M}}(\widehat{N}_{n\mathbb{M}}) = N_n}
\end{array}$$

$$\begin{array}{c}
\frac{\Gamma \vdash a : \mathbb{M}}{\Gamma \vdash b : T_{\mathbb{M}}(a) \rightarrow \mathbb{M}} \quad \mathbb{M}\text{in}_{\Pi} \quad \frac{\Gamma \vdash a : \mathbb{M}}{\Gamma \vdash b : T_{\mathbb{M}}(a) \rightarrow \mathbb{M}} \\
\hline
\Gamma \vdash \widehat{\Pi}_{\mathbb{M}}(a, b) : \mathbb{M} \quad \Gamma \vdash T_{\mathbb{M}}(\widehat{\Pi}_{\mathbb{M}}(a, b)) = \Pi_{(x:T_{\mathbb{M}}(a))}T_{\mathbb{M}}(b \ x)
\end{array}$$

$$\begin{array}{c}
\frac{\Gamma \vdash a : \mathbb{M}}{\Gamma \vdash b : T_{\mathbb{M}}(a) \rightarrow \mathbb{M}} \quad \mathbb{M}\text{in}_{\Sigma} \quad \frac{\Gamma \vdash a : \mathbb{M}}{\Gamma \vdash b : T_{\mathbb{M}}(a) \rightarrow \mathbb{M}} \\
\hline
\Gamma \vdash \widehat{\Sigma}_{\mathbb{M}}(a, b) : \mathbb{M} \quad \Gamma \vdash T_{\mathbb{M}}(\widehat{\Sigma}_{\mathbb{M}}(a, b)) = \Sigma_{(x:T_{\mathbb{M}}(a))}T_{\mathbb{M}}(b \ x)
\end{array}$$

$$\begin{array}{c}
\frac{\Gamma \vdash a : \mathbb{M} \quad \Gamma \vdash b : \mathbb{M}}{\Gamma \vdash a \hat{+}_{\mathbb{M}} b : \mathbb{M}} \text{Min}_+ \quad \frac{\Gamma \vdash a : \mathbb{M} \quad \Gamma \vdash b : \mathbb{M}}{\Gamma \vdash T_{\mathbb{M}}(a \hat{+}_{\mathbb{M}} b) = T_{\mathbb{M}}(a) + T_{\mathbb{M}}(b)} \\
\\
\frac{\Gamma \vdash a : \mathbb{M} \quad \Gamma \vdash b : T_{\mathbb{M}}(a) \rightarrow \mathbb{M}}{\Gamma \vdash \widehat{W}_{\mathbb{M}}(a, b) : \mathbb{M}} \text{Min}_W \quad \frac{\Gamma \vdash a : \mathbb{M} \quad \Gamma \vdash b : T_{\mathbb{M}}(a) \rightarrow \mathbb{M}}{\Gamma \vdash T_{\mathbb{M}}(\widehat{W}_{\mathbb{M}}(a, b)) = W_{(x:T_{\mathbb{M}}(a))} T_{\mathbb{M}}(b x)} \\
\\
\frac{\Gamma \vdash a : \mathbb{M} \quad \Gamma \vdash b : T_{\mathbb{M}}(a) \quad \Gamma \vdash c : T_{\mathbb{M}}(a)}{\Gamma \vdash \widehat{\text{Id}}_{\mathbb{M}}(a, b, c) : \mathbb{M}} \text{Min}_{\text{Id}} \quad \frac{\Gamma \vdash a : \mathbb{M} \quad \Gamma \vdash b : T_{\mathbb{M}}(a) \quad \Gamma \vdash c : T_{\mathbb{M}}(a)}{\Gamma \vdash T_{\mathbb{M}}(\widehat{\text{Id}}_{\mathbb{M}}(a, b, c)) = \text{Id}(T_{\mathbb{M}}(a), b, c)}
\end{array}$$

The rules concerning the codes of non-Mahlo universes are as follows:

$$\begin{array}{c}
\frac{\Gamma \vdash f : \Sigma_{(x:\mathbb{M})}(T_{\mathbb{M}}(x) \rightarrow \mathbb{M}) \rightarrow \Sigma_{(x:\mathbb{M})}(T_{\mathbb{M}}(x) \rightarrow \mathbb{M})}{\Gamma \vdash \widehat{U}_f : \mathbb{M}} \text{Min}_U \\
\\
\frac{\Gamma \vdash f : \Sigma_{(x:\mathbb{M})}(T_{\mathbb{M}}(x) \rightarrow \mathbb{M}) \rightarrow \Sigma_{(x:\mathbb{M})}(T_{\mathbb{M}}(x) \rightarrow \mathbb{M})}{\Gamma \vdash T_{\mathbb{M}}(\widehat{U}_f) = U_f} T_{\mathbb{M}\text{uni}} \\
\\
\frac{\Gamma \vdash f : \Sigma_{(x:\mathbb{M})}(T_{\mathbb{M}}(x) \rightarrow \mathbb{M}) \rightarrow \Sigma_{(x:\mathbb{M})}(T_{\mathbb{M}}(x) \rightarrow \mathbb{M}) \quad \Gamma \vdash a : U_f}{\Gamma \vdash \widehat{T}_f(a) : \mathbb{M}} \text{Min}_{\widehat{T}}
\end{array}$$

Rules for Non-Mahlo Universes ($U_f, \widehat{T}_f$): below we define the U_f -formation rule $U_f\text{fm}$, the U_f -introduction rules $U_f\text{in}$ and the inference rules which define the decoding function $\widehat{T}_f$ recursively. Congruence rules are defined as in the case of the Mahlo Universe rules.

We denote the assumption $\Gamma \vdash f : \Sigma_{(x:\mathbb{M})}(T_{\mathbb{M}}(x) \rightarrow \mathbb{M}) \rightarrow \Sigma_{(x:\mathbb{M})}(T_{\mathbb{M}}(x) \rightarrow \mathbb{M})$ by $(*)$, and put $T_f(a) := T_{\mathbb{M}}(\widehat{T}_f(a))$.

$$\begin{array}{c}
\frac{(*)}{\Gamma \vdash U_f \text{ type}} U_f\text{fm} \quad \frac{(*)}{\Gamma \vdash \widehat{N}_f : U_f} U_f\text{in}_N \quad \frac{(*)}{\Gamma \vdash \widehat{T}_f(\widehat{N}_f) = \widehat{N}_{\mathbb{M}} : \mathbb{M}} \\
\\
\frac{(*) \quad \Gamma \vdash a : U_f \quad \Gamma \vdash b : T_f(a) \rightarrow U_f}{\Gamma \vdash \widehat{\Pi}_f(a, b) : U_f} U_f\text{in}_{\Pi} \quad \frac{(*) \quad \Gamma \vdash a : U_f \quad \Gamma \vdash b : T_f(a) \rightarrow U_f}{\Gamma \vdash \widehat{T}_f(\widehat{\Pi}_f(a, b)) = \widehat{\Pi}_{\mathbb{M}}(\widehat{T}_f(a), \lambda x. \widehat{T}_f(b x)) : \mathbb{M}} \\
\\
\frac{(*) \quad \Gamma \vdash a : U_f \quad \Gamma \vdash b : T_f(a) \quad \Gamma \vdash c : T_f(a)}{\Gamma \vdash \widehat{\text{Id}}_f(a, b, c) : U_f} U_f\text{in}_{\text{Id}} \quad \frac{(*) \quad \Gamma \vdash a : U_f \quad \Gamma \vdash b : T_f(a) \quad \Gamma \vdash c : T_f(a)}{\Gamma \vdash \widehat{T}_f(\widehat{\text{Id}}_f(a, b, c)) = \widehat{\text{Id}}_{\mathbb{M}}(\widehat{T}_f(a), b, c) : \mathbb{M}}
\end{array}$$

Similarly, we define the introduction and decoding function rules for $\widehat{N}_{nf}, \widehat{\Sigma}_f, \widehat{+}_f, \widehat{W}_f$, but we omit to write down them.

The following rules are the rules for the restriction of f to U_f :

$$\begin{array}{c}
(*) \\
\frac{\Gamma \vdash a : U_f}{\Gamma \vdash b : T_f(a) \rightarrow U_f} U_{f \text{ in } \text{res}^0} \quad \frac{\Gamma \vdash a : U_f \quad \Gamma \vdash b : T_f(a) \rightarrow U_f}{\Gamma \vdash \widehat{T}_f(\text{res}_f^0(a, b)) = p_1(f(\widehat{T}_f(a), \lambda x. \widehat{T}_f(b x))) : \mathbb{M}} \\
\\
(*) \\
\frac{\Gamma \vdash a : U_f \quad \Gamma \vdash b : T_f(a) \rightarrow U_f \quad \Gamma \vdash c : T_f(\text{res}_f^0(a, b))}{\Gamma \vdash \text{res}_f^1(a, b, c) : U_f} U_{f \text{ in } \text{res}^1} \quad \frac{\Gamma \vdash a : U_f \quad \Gamma \vdash b : T_f(a) \rightarrow U_f \quad \Gamma \vdash c : T_f(\text{res}_f^0(a, b))}{\Gamma \vdash \widehat{T}_f(\text{res}_f^1(a, b, c)) = p_2(f(\widehat{T}_f(a), \lambda x. \widehat{T}_f(b x))) c : \mathbb{M}}
\end{array}$$

Note that, in the definition above, we defined subuniverses of the Mahlo Universe $\mathbb{M}$ as *recursive subuniverses* in the sense of [Set08b, Set08a]: the decoding function $\widehat{T}_f$ for the subuniverse U_f was defined recursively.

APPENDIX B. ACCESSIBILITY PREDICATE IN EXTENSIONAL TYPE THEORY

This appendix shows that one can define the accessibility predicate Acc by using W -types if the core type theory of **MLM** is replaced with the extensional one in [ML84]. We denote this extensional variant of **MLM** by **MLM_{ext}**. In fact, the argument below can be repeated in extensional **MLTT** of [ML84] itself. So, in **MLM_{ext}**, one can simulate our interpretation of **Pi-Numbers** without adding Acc to the core type theory. None of the results given by this appendix is used in the other sections of the present paper. For comparison, recall that the core type theory of [RGP98] is the extensional type theory of [ML84] too, and that the interpretation of **Pi-Numbers** in [RGP98] does not use Acc at all.

First, we note that the transitive closure function tc enables to formulate a stronger transfinite induction on $\alpha : \mathbb{V}$ in *intensional MLTT*. Since we have this transfinite induction principle also in extensional **MLTT**, we can use it to define the predicate Acc by using W -types in **MLM_{ext}**.

Lemma B.1 (Transfinite Induction on Transitive Closure, **Link**). *Let F be a type with $\Gamma, \beta : \mathbb{V} \vdash F$ type. We then have a term*

$$\text{TI}^{\text{tc}} : \Pi_{(\alpha : \mathbb{V})} (\forall \gamma \in \alpha_{\text{tc}} F[\gamma/\beta] \rightarrow F[\alpha/\beta]) \rightarrow \Pi_{(\alpha : \mathbb{V})} F[\alpha/\beta].$$

We replace the rules Idin , Idel and Ideq with Id^{ein} , Id^{eel} and Id^{eeq} , respectively.

$$\frac{\Gamma \vdash a = b : A}{\Gamma \vdash r : \text{Id}(A, a, b)} \text{Id}^{\text{ein}} \quad \frac{\Gamma \vdash a : A \quad \Gamma \vdash b : A \quad \Gamma \vdash d : \text{Id}(A, a, b)}{\Gamma \vdash a = b : A} \text{Id}^{\text{eel}} \quad \frac{\Gamma \vdash a : A \quad \Gamma \vdash b : A \quad \Gamma \vdash d : \text{Id}(A, a, b)}{\Gamma \vdash d = r : \text{Id}(A, a, b)} \text{Id}^{\text{eeq}}$$

We call the resulting Id -types the *extensional Id-types*.

As is well known, the η -rule for Π -types with the rules for extensional Id -types implies the principle of function extensionality.

Function Extensionality: for any A, B with $\Gamma, x : A \vdash B$ type,

$$\Gamma \vdash \Pi_{(f : \Pi_{(x : A)} B)} \Pi_{(g : \Pi_{(x : A)} B)} (\Pi_{(x : A)} f x =_B g x \rightarrow f =_{\Pi_{(x : A)} B} g)$$

holds.

We then obtain the propositional induction principle for TI^{tc} , which we state as the lemma below. Note that **Function Extensionality** is the only extensional principle on which the lemma depends; so the lemma holds also in intensional **MLTT** with it.

Lemma B.2 (Propositional Induction Principle for TI^{tc} , **Link**). *Let F be a type with $\Gamma, \beta : \mathbb{V} \vdash F$ type. For any $g : \Pi_{(\alpha:\mathbb{V})}(\forall \gamma \in \alpha_{\text{tc}} F[\gamma/\beta] \rightarrow F[\alpha/\beta])$ and any $\alpha : \mathbb{V}$, we have*

$$\text{TI}^{\text{tc}} g \alpha =_{F[\alpha/\beta]} g \alpha (\lambda x. \text{TI}^{\text{tc}} g (\text{pred } \alpha_{\text{tc}} x)).$$

By the propositional induction principle above, one can derive the following computation rule for TI^{tc} in extensional **MLTT**:

$$\frac{\Gamma, \beta : \mathbb{V} \vdash F \text{ type} \quad \Gamma \vdash g : \Pi_{(\alpha:\mathbb{V})}(\forall \gamma \in \alpha_{\text{tc}} F[\gamma/\beta] \rightarrow F[\alpha/\beta]) \quad \Gamma \vdash \alpha : \mathbb{V}}{\Gamma \vdash \text{TI}^{\text{tc}} g \alpha = g \alpha (\lambda x. \text{TI}^{\text{tc}} g (\text{pred } \alpha_{\text{tc}} x)) : F[\alpha/\beta]}$$

This computation rule enables to define the accessibility predicate Acc . Let F be $\mathbb{M}$, and g_{Acc} be

$$\lambda \alpha. \lambda f. \widehat{\Pi}_{\mathbb{M}}(\text{index } \alpha_{\text{tc}}, \lambda x. f x) : \Pi_{(\alpha:\mathbb{V})}(\forall \beta \in \alpha_{\text{tc}} \mathbb{M} \rightarrow \mathbb{M}),$$

then we define $\text{Acc } \gamma := \text{T}_{\mathbb{M}}(\text{TI}^{\text{tc}} g_{\text{Acc}} \gamma)$.

Next, though we do not use this lemma below, we show in extensional **MLTT** that for every $\alpha : \mathbb{V}$, any two terms of type $\text{Acc } \alpha$ are propositionally equal, hence they are also judgementally equal.

Lemma B.3 (Extensional **MLTT**). *Let α be of type $\mathbb{V}$. We then have $d =_{\text{Acc } \alpha} e$ for any two terms d, e of type $\text{Acc } \alpha$.*

Proof. By transfinite induction on transitive closure (Lemma B.1). The induction hypothesis says that

$$\Pi_{(x:\overline{\alpha_{\text{tc}}})} \Pi_{(d':\text{Acc } (\text{pred } \alpha_{\text{tc}} x))} \Pi_{(e':\text{Acc } (\text{pred } \alpha_{\text{tc}} x))} d' =_{\text{Acc } (\text{pred } \alpha_{\text{tc}} x)} e'$$

holds. Consider any arbitrary $d, e : \text{Acc } \alpha$, and we show that $d =_{\text{Acc } \alpha} e$ holds, where we have $d, e : \forall \beta \in \alpha_{\text{tc}} \text{Acc } \beta$ by the TI^{tc} -computation rule:

$$\begin{aligned} \text{Acc } \alpha &= \text{T}_{\mathbb{M}}(\text{TI}^{\text{tc}} g_{\text{Acc}} \alpha) = \text{T}_{\mathbb{M}}(g_{\text{Acc}} \alpha (\lambda x. \text{TI}^{\text{tc}} g_{\text{Acc}} (\text{pred } \alpha_{\text{tc}} x))) \\ &= \Pi_{(x:\overline{\alpha_{\text{tc}}})} \text{T}_{\mathbb{M}}(\text{TI}^{\text{tc}} g_{\text{Acc}} (\text{pred } \alpha_{\text{tc}} x)) = \Pi_{(x:\overline{\alpha_{\text{tc}}})} \text{Acc } (\text{pred } \alpha_{\text{tc}} x) = \forall \beta \in \alpha_{\text{tc}} \text{Acc } \beta. \end{aligned}$$

By **Function Extensionality**, it suffices to verify that $\Pi_{(x:\overline{\alpha_{\text{tc}}})} d x =_{\text{Acc } (\text{pred } \alpha_{\text{tc}} x)} e x$ holds, but this claim follows from IH. $\square$

In the remainder of this section, we derive the introduction, elimination and computation rules for Acc (cf. Definition 3.13) in extensional **MLTT**.

Proposition B.4 (Extensional **MLTT**). *There are two terms prog and E_{Acc} satisfying the rules below.*

$$\frac{\Gamma \vdash \alpha : \mathbb{V} \quad \Gamma \vdash f : \Pi_{(x:\overline{\alpha_{\text{tc}}})} \text{Acc } (\text{pred } \alpha_{\text{tc}} x)}{\Gamma \vdash \text{prog } f : \text{Acc } \alpha} \text{ in}$$

$$\frac{\Gamma \vdash \alpha : \mathbb{V} \quad \Gamma \vdash t : \text{Acc } \alpha \quad \Gamma, \beta : \mathbb{V}, x : \text{Acc } \beta \vdash C \text{ type} \quad \Gamma \vdash g : \Pi_{(\gamma:\mathbb{V})} \Pi_{(f:\Pi_{(y:\overline{\gamma_{\text{tc}}})} \text{Acc } (\text{pred } \gamma_{\text{tc}} y))} ((\Pi_{(y:\overline{\gamma_{\text{tc}}})} C[\text{pred } \gamma_{\text{tc}} y/\beta, f y/x]) \rightarrow C[\gamma/\beta, \text{prog } f/x])}{\Gamma \vdash \text{E}_{\text{Acc}} \alpha t g : C[\alpha/\beta, t/x]} \text{ el}$$

$$\begin{array}{c}
\Gamma \vdash \alpha : \mathbb{V} \\
\Gamma \vdash f : \Pi_{(y:\overline{\alpha_{tc}})} \text{Acc} (\text{pred } \alpha_{tc} y) \\
\Gamma, \beta : \mathbb{V}, x : \text{Acc } \beta \vdash C \text{ type} \\
\hline
\Gamma \vdash g : \Pi_{(\gamma:\mathbb{V})} \Pi_{(f:\Pi_{(y:\overline{\gamma_{tc}})} \text{Acc} (\text{pred } \gamma_{tc} y))} ((\Pi_{(y:\overline{\gamma_{tc}})} C[\text{pred } \gamma_{tc} y/\beta, f y/x]) \rightarrow C[\gamma/\beta, \text{prog } f/x]) \\
\hline
\Gamma \vdash E_{\text{Acc}} \alpha (\text{prog } f) g = g \alpha f (\lambda y. E_{\text{Acc}} (\text{pred } \alpha_{tc} y) (f y) g) : C[\alpha/\beta, \text{prog } f/x] \quad \text{eq}
\end{array}$$

Proof. As shown in the above proof of Lemma B.3, we have $\text{Acc } \alpha = \Pi_{(x:\overline{\alpha_{tc}})} \text{Acc} (\text{pred } \alpha_{tc} x)$. So we can define **prog** as the identity function $\lambda f.f$ on $\Pi_{(x:\overline{\alpha_{tc}})} \text{Acc} (\text{pred } \alpha_{tc} x)$, and then obtain the introduction rule for **Acc**.

Next, we derive the **Acc**-elimination rule. We first provide a term of type

$$\Pi_{(\alpha:\mathbb{V})} \Pi_{(c:\text{Acc } \alpha)} C[\alpha/\beta, c/x].$$

Defining $F := \lambda \alpha. \Pi_{(c:\text{Acc } \alpha)} C[\alpha/\beta, c/x]$, we use Lemma B.1, namely, transfinite induction on transitive closure to construct a term of this type. Take variables α, c and g' of types

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

$$c : \text{Acc } \alpha \text{ and}$$

$$g' : \Pi_{(\gamma:\mathbb{V})} \Pi_{(f:\Pi_{(y:\overline{\gamma_{tc}})} \text{Acc} (\text{pred } \gamma_{tc} y))} (\Pi_{(y:\overline{\gamma_{tc}})} C[\text{pred } \gamma_{tc} y/\beta, f y/x] \rightarrow C[\gamma/\beta, \text{prog } f/x]),$$

respectively. The induction hypothesis is the following assumption:

$$f' : \forall \gamma \in \alpha_{tc} \Pi_{(d:\text{Acc } \gamma)} C[\gamma/\beta, d/x].$$

Since we have $c : \text{Acc } \alpha = \forall \gamma \in \alpha_{tc} \text{Acc } \gamma$, it follows from IH that

$$\lambda y. f' y (c y) : \Pi_{(y:\overline{\alpha_{tc}})} C[\text{pred } \alpha_{tc} y/\beta, c y/x]$$

holds, and we denote the term $\lambda y. f' y (c y)$ by s . We thus obtain $g' \alpha c s : C[\alpha/\beta, \text{prog } c/x]$, and we also have

$$u := \lambda \alpha. \lambda f'. \lambda c. g' \alpha c s : \Pi_{(\alpha:\mathbb{V})} \Pi_{(f':\forall \gamma \in \alpha_{tc} \Pi_{(d:\text{Acc } \gamma)} C[\gamma/\beta, d/x])} \Pi_{(c:\text{Acc } \alpha)} C[\alpha/\beta, c/x]$$

because **prog** c is judgementally equal to c . It thus follows that

$$\text{TI}^{tc} u : \Pi_{(\alpha:\mathbb{V})} \Pi_{(c:\text{Acc } \alpha)} C[\alpha/\beta, c/x]$$

holds with the judgemental equality

$$\text{TI}^{tc} u \alpha = u \alpha (\lambda x. \text{TI}^{tc} u (\text{pred } \alpha_{tc} x)) : \Pi_{(c:\text{Acc } \alpha)} C[\alpha/\beta, c/x].$$

Define $E_{\text{Acc}} := \lambda \alpha'. \lambda c'. \lambda g'. \text{TI}^{tc} u \alpha' c'$. It is obvious that $E_{\text{Acc}} \alpha t g$ is of type $C[\alpha/\beta, t/x]$, hence we have **Acc**-elimination rule. Moreover, we also obtain the **Acc**-computation rule because the following equations hold judgementally:

$$\begin{aligned}
E_{\text{Acc}} \alpha (\text{prog } f) g &= \text{TI}^{tc} u [g/g'] \alpha (\text{prog } f) = u [g/g'] \alpha (\lambda x. \text{TI}^{tc} u [g/g'] (\text{pred } \alpha_{tc} x)) (\text{prog } f) \\
&= g \alpha (\text{prog } f) (\lambda y. (\lambda x. \text{TI}^{tc} u [g/g'] (\text{pred } \alpha_{tc} x)) y (\text{prog } f y)) \\
&= g \alpha (\text{prog } f) (\lambda y. \text{TI}^{tc} u [g/g'] (\text{pred } \alpha_{tc} y) (\text{prog } f y)) \\
&= g \alpha f (\lambda y. \text{TI}^{tc} u [g/g'] (\text{pred } \alpha_{tc} y) (f y)) \\
&= g \alpha f (\lambda y. (\lambda \alpha'. \lambda c'. \lambda g'. \text{TI}^{tc} u \alpha' c') (\text{pred } \alpha_{tc} y) (f y) g) \\
&= g \alpha f (\lambda y. E_{\text{Acc}} (\text{pred } \alpha_{tc} y) (f y) g),
\end{aligned}$$

where the terms **prog** was defined as an identity function. □