Università degli Studi di Padova

Dipartimento di Matematica “Tullio Levi-Civita”

PhD Program in
Mathematical Sciences

PhD Thesis

Around the Minimalist Foundation: (Co)Induction and Equiconsistency

Supervisors:
Prof. Maria Emilia Maietti
Dott. Samuele Maschio

Candidate:
Pietro Sabelli

Academic Year 2023/2024

Ringraziamenti

Desidero innanzitutto ringraziare i miei relatori Milly e Samuele, per avermi trasmesso la loro conoscenza e la loro inestimabile passione. Ringrazio tutto il gruppo di logica padovano per i preziosi confronti che non hanno mai mancato di arricchirmi: Cipriano, Giovanni, Francesco e Silvia. Infine ringrazio la mia famiglia per l'affettuoso supporto. E Federica, per essermi stata vicina.

Contents

Chapter 1

An Introduction to the Minimalist Foundation

Chapter Abstract

In this preliminary chapter, we will introduce the reader to the Minimalist Foundation, both in its philosophical and technical aspects. In doing so, we derive some essential results which we will employ throughout the later chapters.

1.1 Scope and Methods

As opposed to classical mathematics, which relies on an established standard foundation, namely the axiomatic Zermelo-Fraenkel set theory **ZF** (often extended with the Axiom of Choice), constructive mathematics enjoys a variety of possible foundations in which it can be formalised and developed. They come from a wide range of fields: axiomatic set theory, such as Aczel's constructive set theory **CZF** [AR10]; category theory, such as the internal language of a topos $\mathcal{T}_{\mathbf{Topos}}$ [Mai05]; and type theory, such as Martin-Löf's type theory (intensional **MLTT** [NPS90], or extensional **eMLTT** [Mar84]), Coquand-Paulin's Calculus of Inductive Constructions **CIC** [CP90], and Homotopy Type Theory **HoTT** [Uni13]. Each field and system carries its philosophy, language, techniques and insights. We refer to this state of affairs as *pluralism* in foundations.

Since those foundations often use different, incompatible principles, mathematics performed in one cannot be directly compared with that developed in another. To help in this task and maximise the benefits of pluralism, the Minimalist Foundation **MF**, first conceived by Maietti and Sambin in [MS05] and then fully formalised by Maietti in [Mai09], was introduced to serve as a common core for classical and constructive mathematics: extrinsically, through its *compatibility* with other existing theories in the literature, meaning that definitions, theorems and proofs written in its formalism can be exported soundly in the most relevant foundations, as the ones recalled above; and intrinsically, by the *modularity* of its system, which can be flexibly extended to accommodate one's favourite mathematical style – for example, in this work we will discuss its extension with inductive and coinductive definitions (Chapter 2), with propositional extensionality (Chapter 3), and with classical logic (Chapter 4). Of course, these two aspects are closely related, and in many cases compatibility results are achieved by showing that the theory in question is an extension of the Minimalist Foundation, so that they can also be read as results about its modularity; this is indeed the case of most of the type theories mentioned above [Mai09; CM22], including the internal language of toposes [Mai05], and, less trivially, of axiomatic set theories [MS22].

The design principle allowing the Minimalist Foundation to meet those desired properties, whose applications will be encountered in many occasions throughout this introductory chapter, is expressed by the slogan *minimal in assumptions, maximal in distinctions*. On the one hand, fewer assumptions keep the proof-theoretic strength of the system low and enhance its modularity; on the other hand, since we want our foundation to be expressively rich enough for developing non-trivial mathematics, many different notions have to be introduced primitively.

The first and foremost instance of such principle is that **MF** does not consist of a single theory, but of two distinct theories, corresponding to two different levels of abstraction. More in detail, **MF** consists of:

- an *extensional level*, called **emTT** for *extensional minimal type theory*, which is a calculus close to the ordinary mathematical language and practice, understood as the intended place to formalise and develop predicative constructive mathematics;

- an *intensional level*, called **mTT** for *minimal type theory*, which is a calculus acting as a functional programming language enjoying a Kleene realisability interpretation [TD88];

- an interpretation, called *quotient* or *restore interpretation*, of the extensional level into a quotient model built over the intensional one.

We refer to this situation by saying that **MF** is a *two-level foundation* [Mai09].

The presence of two levels makes it possible, from time to time, to choose the appropriate one depending on the nature of the foundation to be compared. This drastically improves the compatibility of the overall system and actually reveals two distinct common cores: one among intensional theories such as Martin-Löf's type theory and the Calculus of Constructions; and the other among extensional ones, such as axiomatic set theories and the internal language of toposes. What is more, having two levels turns out to be compulsory for a foundation which wants to be compatible with all the different interpretations of constructive mathematics, since, as we recall in Section 1.2, they give rise to conflicting demands that no single theory can fulfil at once.

1.1.1 Formal Topology

Historically, topology has always been a stress test for the formalisation of mathematics; this is especially true for predicative constructive mathematics, in which even the traditional definition of topological space is unacceptable given its use of the Power-set Axiom. Formal Topology is the study of topology in a constructive and predicative setting put forward by P. Martin-Löf and G. Sambin in [Sam87]; such notion was recently enriched to that of *basic topology*, which primitively represents the basic open and closed subsets (see [Sam03]) and whose definition is predicatively sound.

The desire of having a suitable foundation for the development of Formal Topology was the other main impulse for the introduction of the Minimalist Foundation. In turn, Formal Topology witnesses that a notoriously difficult subject for formalisation as that of topology can be carried out in **MF**, and consequently that non-trivial mathematics can be obtained in it with little formalisation effort. Sambin's book [Sam24] is a major witness in the philosophical and technical aspects of this intertwined relation between the formal system and the goal object of its formalisation.

In Chapter 2, we will contribute to the formalisation of Formal Topology by showing a tight correspondence between the (co)inductive generation of basic topologies and other general-purpose schemes of (co)induction.

1.2 Compatibility with foundational approaches

The pluralist vision pursued by the Minimalist Foundation, presented above in terms of compatibility with fully formal foundations, does not end, nor start, with them. General approaches to the foundations of mathematics, independently to their eventual implementations in formal systems, have nevertheless provided a crucial motivation for its introduction.

On a very ground level, this is the case of the broad spectrum of constructive mathematics, as opposed to classical mathematics; and that of predicative mathematics, as opposed to impredicative mathematics. It is clear that, if a foundation wants to be compatible with both pairs of opposing philosophies, it must embrace the firsts, i.e. be constructive and predicative, since everything that can be proved in them can also be proved in the seconds, while the reverse does not apply. In formal terms, a minimum requirement (and some might even consider it sufficient) to be constructivist is to reject

– that is *not to validate* – the Law of the Excluded Middle LEM; thus, the underlying logic of the system must be intuitionistic. An analogous role is played for predicativism by the Power-set Axiom PSA, which must be rejected to prevent circular definitions.

$$\mathsf{LEM}\ \frac{\varphi\ prop}{\varphi \vee \neg\varphi\ true} \qquad \mathsf{PSA}\ \frac{A\ set}{\mathcal{P}(A)\ set}$$

Starting from that, there are different combinations and interpretations in which these philosophies can be delineated. We now recall those having high conceptual relevance for the Minimalist Foundation and for foundations in general.

1.2.1 Computational interpretation of constructivism

The recursive interpretation of constructive mathematics wants every notion to be effective, computable. For intuitionistic logic, this is the case of the Brouwer-Heyting-Kolmogorov interpretation, which regards proofs as algorithms, thus holding in particular the validity of the Axiom of Choice.

$$(\forall x \in A)(\exists y \in B)R(x, y) \Rightarrow (\exists f \in A \to B)(\forall x \in A)R(x, f(x)) \qquad (\mathsf{AC})$$

On the other hand, in the recursive interpretation of arithmetic, every function is computable, that is programmable as a Turing machine; such view validates the formal Church-Turing thesis

$$(\forall f \in \mathbb{N} \to \mathbb{N})(\exists e \in \mathbb{N})(\forall x \in \mathbb{N})(\exists y \in \mathbb{N})(\mathsf{T}(e, x, y) \wedge \mathsf{U}(y) =_{\mathbb{N}} f(x)) \quad (\mathsf{CT})$$

where $\mathsf{T}(e, x, y)$ is the Kleene predicate expressing that y is the encoding of the computation history of the computable function encoded by e on input x, and $\mathsf{U}(y) \in \mathbb{N}$ is the output of such computation.

In [Mai09], the authors proposed the consistency of a theory with $\mathsf{AC}+\mathsf{CT}$ as a formal criterion of compatibility with the computable interpretation of constructivism. More in detail, they named *proofs-as-programs theories* those enjoying program extraction from their proofs through a realisability interpretation à la Kleene [TD88] validating $\mathsf{AC} + \mathsf{CT}$.

This strict requirement highly conflicts with the usual mathematical practice and clearly shows the necessity of a two-level structured foundation. In fact, the pair of axioms $\mathsf{AC} + \mathsf{CT}$ is incompatible both with LEM, and with the principle of function extensionality

$$(\forall f, g \in A \to B)((\forall x \in A) f(x) =_B g(x) \ \Rightarrow \ f =_{A \to B} g) \qquad (\mathsf{funext})$$

This has been proved in [TD88] within the basic setting of intuitionistic arithmetic of finite types $\mathbf{HA}^\omega$.

$$\mathbf{HA}^\omega + \mathsf{AC} + \mathsf{CT} + \mathsf{funext} \vdash 0 = 1$$
$$\mathbf{HA}^\omega + \mathsf{AC} + \mathsf{CT} + \mathsf{LEM} \vdash 0 = 1$$

These behaviours rule out from being proof-as-program theories not only classical foundations but also extensional constructive theories such as **CZF** and $\mathcal{T}_{\mathbf{Topos}}$, and even intensional theories with extensional features such as **HoTT**.

Moreover, even among full intensional type theories, it is still an open problem to determine whether the impredicative ones such as **CIC** can be considered proofs-as-programs theories; until recently, the same question was open for intensional theories satisfying the ξ-equality rule of lambda abstraction

$$\xi \ \frac{t(x) = s(x) \in B(x) \ [x \in A]}{(\lambda x \in A)t(x) = (\lambda x \in A)s(x) \in (\Pi x \in A)B(x)}$$

such as the standard versions of **MLTT**; this was answered positively in [Péd24].

The intensional level **mTT** of **MF** (which indeed does not assume the above ξ-rule) and a version of **MLTT** without the ξ-rule were proved in [IMMS18] to be proofs-as-programs theories. Here, it is perhaps worth stressing that **mTT** does not validate neither AC nor CT, and in fact it is even consistent with both their negations.

1.2.2 Bishop constructivism

Among the possible interpretations of constructive mathematics, the one put forward by Bishop in his treatment of real analysis [BB12] is notably one of the most oriented towards a pluralist view, which made it a natural benchmark for the design of **MF**. Bishop's main aim was to develop constructive mathematics not in opposition to classical mathematics but as a generalisation of the latter, while, at the same time, keeping the possibility of extracting computational content from proofs.

Many systems, among which **CZF**, **MLTT**, and **HoTT**, were proposed as possible formalisation of Bishop mathematics, and the Minimalist Foundation was introduced as a common core among them. However, **MF** does not claim to exactly capture Bishop mathematics; one of the main reasons is that, while Bishop mathematics is compatible with classical, impredicative mathematics such as that formalised by **ZF**, it is nevertheless incompatible with classical predicativism, as we recall in the next subsection.

For a comprehensive discussion of the relationship between **MF** and Bishop's mathematics see [MS23b; CM24].

1.2.3 Classical predicativism à la Weyl

In *Das Kontinuum* [Wey18], Hermann Weyl advocated – and proved possible – the development of real analysis in a classical and predicative fashion. One of his main tenets was that to avoid any form of circularity, the continuum, paradigmatically embodied by real numbers, should not be considered at the same level as sets, such as that of natural numbers. In his words, *not the relationship of an element to a set, but of a part to a whole ought to be taken as a basis for the analysis of a continuum.* Real numbers, and, more generally, the collection of subsets of a given set were something whose existence was admissible, but not as an *extensionally determinate domain*; in particular, no new set could be formed by quantification over them. A similar treatment was applied to the *collection* of functional relations – i.e. subsets of pairs $R \in \mathcal{P}(A \times B)$ satisfying the functionality condition

$$(\forall x \in A)(\exists! y \in B)R(x, y)$$

– as opposed to the *set* of *rule-based functions*.

As the reader will have the opportunity to acknowledge, the design and terminology of the Minimalist Foundation closely resemble the ideas foreshadowed in Weyl's treatment. In this elaborate, we will present metamathematical results allowing us to upgrade such correspondence to a claim of compatibility. In particular, in Chapter 4 we will prove that the extensional level **emTT** of **MF** is equiconsistent with its classical version; from this, we will deduce that the classical version of **emTT** is predicative and, in particular, that the collections of real numbers à la Dedekind and number-theoretic functional relations do not form sets in it; in turn, these facts philosophically allow us to see it as a formal foundation for classical predicativism à la Weyl, and compatibility with it is achieved.

We remark that compatibility with classical predicative mathematics à la Weyl is rather rare among constructive predicative foundations, since most of them, such as **CZF**, **MLTT**, or **HoTT**, become impredicative when they assume classical logic – in particular, the first one notoriously becomes **ZF**. The source of their incompatibility is to be traced to the Axiom of Unique Choice

$$(\forall x \in A)(\exists! y \in B)\varphi(x,y) \Rightarrow (\exists f \in A \to B)(\forall x \in A)\varphi(x, f(x)) \qquad (\mathsf{AC!})$$

This seemingly innocent axiom is, quoting M. E. Maietti, "a bomb ready to explode"; where the thing exploded is the proof-theoretic strength of the system, and the trigger is precisely LEM. This behaviour can be observed again in the simple case of intuitionistic arithmetic of finite types

$$\mathbf{HA}^{\omega} + \mathsf{AC!} + \mathsf{LEM} \vdash \mathsf{CA}$$

where impredicativity is encoded through the Comprehension Axiom

$$(\exists f \in \mathbb{N} \to \mathbb{N})(\forall x \in \mathbb{N})(\varphi(x) \Leftrightarrow f(x) = 1) \qquad (\mathsf{CA})$$

In turn, $\mathbf{HA}^{\omega}$ + AC! can be easily interpreted in all the theories aforementioned, but not in **MF**, both of whose levels have been proved in [Mai17] not to validate AC!.

The results recalled in this subsection and their refinements are proved and discussed in [CM24], together with an in-depth analysis of the relations between **MF** and Weyl's mathematics.

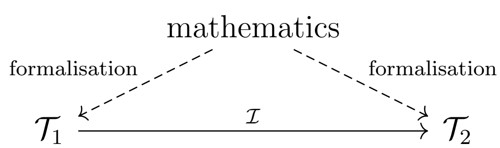
To sum up, **MF** is compatible with the computational interpretation of constructivism through its intensional level **mTT**; it is compatible with the various systems proposed for formalising Bishop mathematics, and, in particular, its extensional level **emTT** sits at the crossroad between Bishop constructivism and Weyl predicativism.

1.3 Towards a formal notion of compatibility

In the previous sections, we often referred to the notion of compatibility between theories, suggesting that a theory is compatible with another one if the former can be interpreted in the latter while preserving the meaning of its logical and set-theoretical entities; and we presented this property as the main raison d'être of the Minimalist Foundation, stating its compatibility with both formal foundations and with general foundational approaches for mathematics. In this section we elaborate on the notion of compatibility between theories based on its definition in [CM22].

First of all, it is clear that our account of this notion relative to pre-formal foundational approaches, by their very nature, could not be ever stated fully formally – although, on a case-by-case basis, it can be endorsed by mathematical results, such as those mentioned in Section 1.2. Therefore, in the following we restrict our attention to the case of formal theories.

As a first approximation, we say that a theory $\mathcal{T}_1$ is *compatible* with another theory $\mathcal{T}_2$ if there exists an interpretation $\mathcal{I} : \mathcal{T}_1 \to \mathcal{T}_2$ which preserves the intended way of formalising mathematics in them, and in this case $\mathcal{I}$ is called a *compatible interpretation*. Pictorially, we are asking for the following diagram to commute.



Clearly, this definition is still fairly philosophical; nevertheless, in most practical cases, the intended meaning of the dashed arrow in the diagram above is evident and unambiguous, as the following examples will hopefully show.

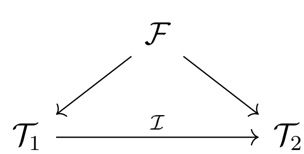
We will also be interested in the particular case where the theory $\mathcal{T}_2$ is an *extension* of the theory $\mathcal{T}_1$, meaning that the former is obtained by extending the latter with new axioms or inference rules. In such a case, if the compatible interpretation $\mathcal{I}$ is the identity interpretation – that is, the interpretation which sends each expression of the source theory to itself – we say that $\mathcal{T}_2$ is a *compatible extension* of $\mathcal{T}_1$.

Examples of compatible interpretations are the standard model of arithmetic in axiomatic set theory, which interprets Peano's arithmetic into classical set theory $\mathbf{PA} \to \mathbf{ZF}$; or the identical interpretation of the constructive version of Zermelo–Fraenkel set theory into the standard, classical one $\mathbf{CZF} \hookrightarrow \mathbf{ZF}$. To test their compatibility, consider the pre-formal notion of *the natural number two*; it is formally expressed in **PA** as the term $\mathsf{SS0}$; while, both in **CZF** and **ZF**, as the set $\{\emptyset, \{\emptyset\}\}$. Both interpretations respect such formalisations, and the same goes for all other terms and propositions of arithmetic.

As non-examples, consider Gödel's double negation translation of classical Peano Arithmetic into intuitionistc Heything Arithmetic $\mathbf{PA} \to \mathbf{HA}$, or the identical interpretation of Martin-Löf's type theory into Homotopy Type Theory $\mathbf{MLTT} \hookrightarrow \mathbf{HoTT}$. In both cases, an existential statement formalised in the domain theory is sent to something that does not correspond at all to the formalisation of an existential statement in the target theory; e.g. the pre-formal proposition *there exists a natural number equal to itself* is formalised as the type $(\Sigma x \in \mathbb{N})\mathsf{Id}(\mathbb{N}, x, x)$ in **MLTT**, but as its propositional truncation $\|(\Sigma x \in \mathbb{N})\mathsf{Id}(\mathbb{N}, x, x)\|$ in **HoTT**; on the other hand, it is formalised both in **PA** and **HA** as the formula $\exists x(x = x)$, which is however sent by the double-negation translation to $\neg\neg\exists x(x = x)$.

Notice that the above examples and non-examples showed that the identical interpretation of a theory into an extension of it can or cannot be a compatible interpretation: the theory **PA** is a compatible extension of **HA**, while **HoTT** is not a compatible extension of **MLTT**.

As a second step, we can refine the above definition of compatibility by replacing pre-formal mathematics with a chosen theory $\mathcal{F}$ acting as a lingua franca between $\mathcal{T}_1$ and $\mathcal{T}_2$. When comparing foundations for constructive mathematics, this role can be played by theories such as $\mathbf{HA}^\omega$ in [TD88], or fragments of intuitionistic second-order arithmetic, since these will certainly possess an intended interpretation in both $\mathcal{T}_1$ and $\mathcal{T}_2$ if these latter are proposed as sufficiently rich theories for the foundations of mathematics. Then, given two interpretations $\mathcal{F} \to \mathcal{T}_1$ and $\mathcal{F} \to \mathcal{T}_2$, we say that $\mathcal{T}_1$ *is compatible with* $\mathcal{T}_2$ *relatively to the given interpretations of* $\mathcal{F}$ if there exists an interpretation $\mathcal{I} : \mathcal{T}_1 \to \mathcal{T}_2$ which makes the following triangle commute.



We stress once more that while the general notion of compatibility described here is not yet fully formal (simply because the one of *theory* itself is not), it can already serve well as a reference in each specific case. Therefore, in the rest of the discussion, and especially in Section 1.8, we will rely on this last notion with respect to $\mathbf{HA}^\omega$ to state the compatibility of **MF** with other foundations.

Finally, we will often resort to two auxiliary concepts introduced hereafter in proving compatibility results between theories.

Constructors encodings We will often find ourselves in the situation of having to prove that a given dependent type theory is expressive enough to encode a given type constructor. To this end, we introduce the following notion, which will be often referred to in Section 1.8 and throughout Chapter 2.

If $\mathcal{T}$ is a dependent type theory and C is a type constructor – intended as the sets of its rules defined in the language of $\mathcal{T}$ – we say that $\mathcal{T}$ *encodes* C if each new symbol appearing in C can be interpreted in $\mathcal{T}$ in such a way that all the rules of C are valid under this interpretation. Moreover, given two type constructors C and D, we say that they are *mutually encodable over* $\mathcal{T}$ if $\mathcal{T} + \mathsf{C}$ encodes D and $\mathcal{T} + \mathsf{D}$ encodes C.

As an example, consider the type of natural numbers $\mathbb{N}$ primitively defined as an inductive type in the usual way; it is encodable as follows in any version of **MLTT** having the singleton type N_1 and the list constructor List.

$$\begin{aligned} \mathbb{N} &:\equiv \mathsf{List}(\mathsf{N}_1) \\ 0 &:\equiv \epsilon \\ \mathsf{succ}(n) &:\equiv \mathsf{cons}(n, \star) \\ \mathsf{El}_{\mathbb{N}}(n, b, (x, z).c) &:\equiv \mathsf{El}_{\mathsf{List}}(n, b, (x, y, z).c) \end{aligned}$$

Alternatively, consider the impredicative encoding of the following logical constructors already possible in the base Calculus of Constructions.

$$\begin{aligned} \bot &:\equiv (\forall \psi \in \mathsf{Prop})\psi \\ \varphi \Rightarrow \psi &:\equiv (\forall x \in \varphi)\psi \\ \varphi \wedge \psi &:\equiv (\forall \phi \in \mathsf{Prop})((\varphi \Rightarrow \psi \Rightarrow \phi) \Rightarrow \phi) \\ \varphi \vee \psi &:\equiv (\forall \phi \in \mathsf{Prop})((\varphi \Rightarrow \phi) \Rightarrow (\psi \Rightarrow \phi) \Rightarrow \phi) \\ (\exists x \in A)\varphi(x) &:\equiv (\forall \psi \in \mathsf{Prop})((\forall x \in A)(\varphi(x) \Rightarrow \psi) \Rightarrow \psi) \\ \mathsf{Id}(A, a, b) &:\equiv (\forall P \in A \to \mathsf{Prop})(P(a) \Rightarrow P(b)) \end{aligned}$$

All the above encodings satisfy the intuitionistic rules of the corresponding logical constructor. In particular, the last encoding satisfies the rules of Leibniz's equality.

Equivalence of theories Usually, we want to consider theories independently of their specific presentation. For this reason, we introduce the following notion of equivalence between theories.

Given a theory $\mathcal{T}$, proposed as a foundation for mathematics, we can consider the category $\mathbf{Set}_{\mathcal{T}}$ of sets and functions formalised in such foundation. We then say that two foundational theories $\mathcal{T}_1$ and $\mathcal{T}_1$ are *equivalent* if the corresponding categories of sets $\mathbf{Set}_{\mathcal{T}_1}$ and $\mathbf{Set}_{\mathcal{T}_2}$ are equivalent in the sense of category theory.

1.4 Levels of abstraction

There is a long-dated and pervasive tension in type theory between intensional and extensional representations of mathematical concepts. Intuitively, mathematicians want to be able to consider mathematical objects independently of their particular presentations, while computer scientists are always bound to these. This has theoretical consequences: extensional type theories are closer to ordinary mathematical language and practice, but their computational properties are inferior, most notably they lack decidable type checking, which makes them harder to implement in proof assistants; conversely, intensional type theories generally possess valuable computational properties such as normalisation, but doing mathematics in them can be limiting and cumbersome.

In the literature, we can distinguish two main opposite ways to tackle this problem. In one direction, one wants to design intensional type theories satisfying desired extensional features in such a way as not to destroy their computational properties, so that one single theory can serve all purposes; this is the spirit for example of Homotopy Type Theory and derived theories such as Cubical Type Theory [SA21]; another example comes from proof assistants such as Agda or Coq, which by default implement a version of Martin-Löf's type theory and the Calculus of Constructions validating η-equality rules for Π- and Σ-types while retaining the decidability of type checking. The other approach, pursued for example in [SV98] and [Hof95], tries to model extensional type theories using intensional ones; according to this paradigm, computer scientists should always be able to interpret in an intensional theory what mathematicians work out extensionally.

Of the two approaches described above, the Minimalist Foundation, being a two-level foundation, follows the latter. Indeed, its extensional level **emTT** is a theory ideally thought for pen-and-paper mathematics, with all the desired extensional features a mathematician would wish for, among which equality reflection, function extensionality, subset extensionality, effective quotients, and proof-irrelevance – moreover, in Chapter 3, we will show that one can also conservatively assume propositional extensionality. Constructions and proofs performed in the extensional level can then be interpreted in the intensional one **mTT**, which is thought as a functional programming language (with no extensional features whatsoever) for computer-aided formalisation of mathematics.

More specifically, the two levels **emTT** and **mTT** interact according to Sambin's *forget-restore* design principle [SV98], which states that extensional concepts should be obtained from intensional ones by abstracting – that is forgetting – irrelevant computational information, in such a way that it should always be possible, knowing how an extensional object has been constructed, to restore the missing data and implement it back by intensional means. Both the forget and the restore part were implemented in [Mai09]; the former, by designing **emTT** as a theory obtained by strengthening the equalities of types and terms of **mTT** and introducing a quotient set constructor; the latter, by interpreting the extensional level into a quotient model built over the intensional one. Technically, such implementation acts with an *hide externally-reveal internally* mechanism: it hides the computational content of extensional judgements in their derivations, which belong to the meta-theory; and it defines an interpretation able to read these latter, and to reveal the computational content internally, as judgements of the intensional level. A key example of retrieved computational content is the case of proof-terms. In **emTT**, proof-terms are abstracted away by imposing a single canonical term true, which, when it appears in a judgement of the form $\mathsf{true} \in \varphi$, does not convey any information about the proof used to assert the truth of the statement φ. Nevertheless, a fully explicit proof-term can still be recovered in **mTT** by reading a derivation of the judgement $\mathsf{true} \in \varphi$.

The quotient interpretation of **emTT** into **mTT** is a cornerstone of the Minimalist Foundation, and we think a simple hands-on example could be useful to let the reader appreciate its basic functioning.

Example 1.4.1. Consider the following (derivable) judgement of **emTT**, asserting the truth of a proposition in modular arithmetic

$$\mathbf{emTT} \vdash \mathsf{true} \in (\exists z \in \mathbb{N}/{\equiv_3})\, z^2 =_{\mathbb{N}/\equiv_3} [1]$$

and suppose we know the following specific derivation of the above judgement (where we have omitted for simplicity some branches).

$$
\textsf{I-}\exists\ \dfrac{\textsf{I-Q}\ \dfrac{\mathbb{N}/{\equiv_3}\ \mathit{set} \quad 2 \in \mathbb{N}}{[2] \in \mathbb{N}/{\equiv_3}} \qquad \dfrac{\dfrac{1 \in \mathbb{N} \quad \mathsf{true} \in 2^2 =_{\mathbb{N}} 3 \cdot 1 + 1}{\mathsf{true} \in (\exists k \in \mathbb{N}) 2^2 =_{\mathbb{N}} 3k + 1}\ \textsf{I-}\exists}{\mathsf{true} \in [2]^2 =_{\mathbb{N}/\equiv_3} [1]}\ \textsf{eq-Q}}{\mathsf{true} \in (\exists z \in \mathbb{N}/{\equiv_3}) z^2 =_{\mathbb{N}/\equiv_3} 1}
$$

The quotient interpretation interprets the extensional quotient set $\mathbb{N}/{\equiv_3}$ as the intensional setoid $(\mathbb{N}, \equiv_3)$, and, accordingly, the extensional equality predicate $[n] =_{\mathbb{N}/\equiv_3} [m]$ as the intensional relation $(\exists k \in \mathbb{N}) n =_{\mathbb{N}} 3k + m$; finally, it turns the whole derivation above into the following decidable judgement of **mTT**.

$$
\mathbf{mTT} \vdash \langle 2, 1, \mathsf{id}(4) \rangle \in (\exists n \in \mathbb{N})(\exists k \in \mathbb{N}) n^2 =_{\mathbb{N}} 3k + 1
$$

Notice that now the proof-term is completely explicit, allowing us in particular to retrieve a witness to the existential statement.

In turn, after having interpreted an extensional derivation of **emTT** as an intensional judgement of **mTT**, we can extract from the latter a program, in the form of a Gödel code. This can be done by following the interpretation in [IMMS18] of **mTT** into a realisability model built in Feferman's predicative arithmetic of non-iterative fixpoints $\widehat{\mathbf{ID}}_1$ [Fef82]. We regard such a realisability model as a third level of abstraction of **MF**, the lowest, whose language is akin to a machine code.

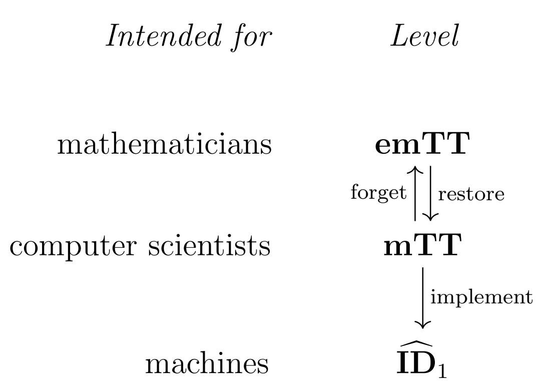


Figure 1.1: Three levels of abstraction of **MF**

Figure 1.1 depicts the three levels of abstraction of the Minimalist Foundation, their conceptual connections (each corresponding to a formal interpretation), and their intended users. Notice that, in principle, nothing prevents mathematicians from working and developing mathematics at the intensional or the realisability level – however, they must be prepared to live in the so-called *setoid hell*, or, even worse, that of arithmetic encodings. Likewise, the extensional and the realisability levels can be interpreted as programming languages too, but the computational behaviour of the first is poor, while the second is low-level.

Remark 1.4.2. As proven in [CM22], **HoTT** has the peculiar property of being able of interpreting in a compatible way both the intensional and the extensional level of **MF**. We expect this is also the case for other type theories that follow the same approach described at the beginning of this section of fitting in between full intensionality and full extensionality.

Moreover, we recall that the realisability interpretation of **mTT** in $\widehat{\mathbf{ID}}_1$ factors through a version of **MLTT**. This is expected, since **MLTT** can be read as a functional programming language primitively enforcing the Curry-Howard isomorphism inherent to the computational interpretation of constructivism (see the discussion in Subsection 1.2.1). In this sense **HoTT**, being an extension of **MLTT**, is a theory capable of hosting even the realisability level of **MF**.

Recall from Section 1.1 that a key property of **MF** is its modularity. We close this section with a brief comment on how this feature should interact with its threefold levels of abstraction.

A constructive mathematician working with the Minimalist Foundation might want additional features to develop a specific topic – we remark here that this scenario is perfectly expected, and even desired by a minimalist foundation. To this end, it is not enough to extend the mathematician's level, namely **emTT**; to keep a realisability interpretation assuring that the proofs still possess a computational content, the whole system of **MF** must be extended to support the additional desired features. In practice, when we want to add a new type constructor, this requires adding an extensional version of it to **emTT**, an intensional version of it to **mTT**, and checking that both the quotient model and the realisability interpretation can be upgraded to support it.

This is exactly what happened in the case of (co)inductive methods of Formal Topology [CSSV03; Sam03; Sam24]: the three levels of **MF** have been extended in [MMR21] to support an inductive topological constructor, and in [MMR22] to support a dual, coinductive counterpart; Chapter 2 will be devoted to analyse such extensions under a different perspective, and to establish new compatibility results for the extended version in the spirit of [Mai19]. Another example in this direction is the work in [Bre15], which extends the quotient interpretation to support the additional presence of W-types in both **emTT** and **mTT**.

1.5 Types variety

In foundations, there are two important kinds of distinctions between entities. The first is between logical and set-theoretical ones; in axiomatic set theories, which are defined on top of predicate logic, or in logic-enriched type theories, which are akin to multi-sorted logic, this separation is clear-cut and built-in; however, when logic is part of a type theory, it can differ from the whole of it with different degrees: the Calculus of Constructions treats logic as a primitively distinct portion of the type theory; both in the internal language of a topos and Homotopy Type Theory, logic is identified through a characterisation, namely that of mono-types and h-propositions, respectively; finally, in the extreme case of Martin-Löf's type theory, the distinction disappears, since its type theory fully identifies logic and set theory.

The second recurring distinction in foundations is on what is usually called the *size* of a collection. We remark that a priori this has no relation with the notion of cardinality from set theory; constructively, the size measured is not that of the number of elements in a given set or type, rather it refers to the complexity of its construction. In this sense, axiomatic set theories distinguish between sets and classes, while type theories usually distinguish between small and large types relative to a given universe. The specific way this distinction is enforced in each theory determines its degree of impredicativity, that is, the possibility of giving circular definitions; when such power is unrestricted, it often leads to inconsistency (and this is how historically the need for such distinction was first recognised); examples of fully predicative theories, that is theories in which no amount of circularity is allowed, are **CZF** and **MLTT**; controlled forms of impredicativity are nevertheless common also in constructive mathematics, such as in the Intuitionistic Zermelo-Frankel set theory **IZF**, **CIC** and $\mathcal{T}_{\mathbf{Topos}}$.

The Minimalist Foundation embraces both of the above distinctions to the maximum by identifying, in both of its levels **emTT** and **mTT**, four kinds of types: *small propositions*, *propositions*, *sets* and *collections* – denoted respectively $prop_s$, $prop$, set and col. Formally, they are rendered as the following typehood judgements.

$$\varphi\ prop_s \qquad \varphi\ prop \qquad A\ set \qquad A\ col$$

Sets are particular collections, just as small propositions are particular propositions; moreover, a proposition (respectively, a small proposition) is identified with the collection (respectively, the set) of its proofs. Eventually, we have the square of inclusions between kinds of types depicted in Figure 1.2. The inclusions are formalised through the following rules.

$$\textsf{prop}_\textsf{s}\textsf{-into-prop}\ \frac{\varphi\ prop_s}{\varphi\ prop} \qquad \textsf{prop}_\textsf{s}\textsf{-into-set}\ \frac{\varphi\ prop_s}{\varphi\ set}$$

$$\textsf{prop-into-col}\ \frac{\varphi\ prop}{\varphi\ col} \qquad \textsf{set-into-col}\ \frac{A\ set}{A\ col}$$

The distinction between logic and set theory is clear: propositions are statements that can be judged true, whereas sets and collections are domains over which quantification can be performed and whose elements can be compared for equality. On the other hand, the *size* distinction is deployed in the Minimalist Foundation with a peculiar qualitative stance to it. Contrary to a plain quantitative size label attached to an object, it reflects the philosophical distinction between effective and open-ended domains. Intuitively, a set is an inductively generated domain, such as the natural numbers $\mathbb{N}$, whose canonical elements are fixed in advance and are not subject to possible extensions of the ambient theory. On the contrary, the elements of a collection could be potentially ever undetermined, consider for example the *subsets of* natural numbers $\mathcal{P}(\mathbb{N})$, which may increase in number as soon as the theory becomes more expressive. Accordingly, small propositions are those propositions which have quantifiers and equality predicates restricted to sets.

The distinction between effective and open-ended domains avoids impredicative definitions; in particular, in **emTT** one can form the power of a set, but the resulting type is a collection.

$$\frac{A\ set}{\mathcal{P}(A)\ col}$$

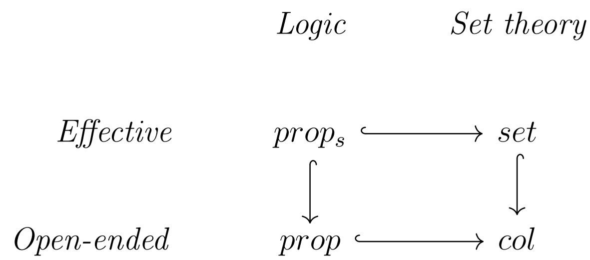


Figure 1.2: Kinds in the Minimalist Foundation

On the other hand, treating logic primitively, as in the Calculus of Constructions, allows to control the information flow from proofs to programs and prevent choice principles from being validated. In **MF**, the compartmentalisation of logic is achieved through an additional refinement: that of allowing the elimination rules of inductive propositional constructors to act only towards propositions. The crucial consequences of this design are best seen in the case of the existential quantifier. Consider the elimination rule of the existential quantifier, and compare it to the one of dependent sum, which can instead acts towards dependent collections.

$$\mathsf{E}\text{-}\exists\ \frac{\begin{array}{l}\psi\ prop \\ c \in (\exists x \in A)\varphi(x) \quad m(x,y) \in \psi\ [x \in A, y \in \varphi(x)]\end{array}}{\mathsf{El}_\exists(c,(x,y).m) \in \psi}$$

$$\mathsf{E}\text{-}\Sigma\ \frac{\begin{array}{l}M(z)\ col\ [z \in (\Sigma x \in A)B(x)] \\ c \in (\Sigma x \in A)B(x) \quad m(x,y) \in M(\langle x,y\rangle)\ [x \in A, y \in B(x)]\end{array}}{\mathsf{El}_\Sigma(c,(x,y).m) \in M(c)}$$

Notice that it is always possible to extract the first component of a pair $c \in (\Sigma x \in A)B(x)$ as the term $\pi_1(c) :\equiv \mathsf{El}_\Sigma(c,(x,y).x) \in A$; however, the same idea cannot (always) be used for extracting the witness of an existential proof $p \in (\exists x \in A)\varphi(x)$, since the term $\mathsf{El}_\exists(p,(x,y).x)$ is bad typed whenever A is not a proposition.

Finally, the main consequence of having logic included in the type theory is that it allows to form new collections by comprehension with a proposition, as well as new sets by comprehension with a small proposition. This is achieved through the dependent sum constructor, as shown in the following derived rules.

$$\frac{A\ col \quad \varphi(x)\ prop\ [x \in A]}{(\Sigma x \in A)\varphi(x)\ col} \qquad \frac{A\ set \quad \varphi(x)\ prop_s\ [x \in A]}{(\Sigma x \in A)\varphi(x)\ set}$$

1.5.1 Comparison with other foundations

It could appear somewhat paradoxical, but, despite constructive mathematics being defined by the fact that it takes existence seriously, in most of its type-theoretical foundations the existential quantifier is not a primitive concept, with decisive proof-theoretic consequences. Here, we argue that this phenomenon is always the result of an extra assumption imposed on the Minimalist Foundation's square 1.2, resulting in the deletion of some distinction.

More generally, as discussed in the beginning of this section, variants of the **MF** square can be found in almost all foundations; the analysis of their relationship is often decisive in obtaining compatibility results. We shall now present the most important cases.

Second-order arithmetic The above examples of $\mathbb{N}$ and $\mathcal{P}(\mathbb{N})$ for set and collection, respectively, are not casual. The **MF** square 1.2 can be interpreted as an enhancement of the usual distinctions in (fragments of) second-order arithmetic between the two sorts $\mathbb{N}$ and $\mathcal{P}(\mathbb{N})$, and between arbitrary formulas and *arithmetical* ones, which are precisely those formulas in which quantification is restricted to $\mathbb{N}$ (see Figure 1.3). In particular, the intended interpretation of the *arithmetic comprehension axiom* **ACA** [Sim09] fragment of second-order arithmetic into the Minimalist Foundation is a direct formalisation of this idea.

	Formulas	*Sorts*
First-order	*arithmetical* $\hookdownarrow$	$\mathbb{N}$
Second-order	*arbitrary*	$\mathcal{P}(\mathbb{N})$

Figure 1.3: Entities in second-order arithmentic

Martin-Löf's type theory In **MLTT**, where the propositions-as-sets paradigm is enforced, the square of **MF** is horizontally collapsed (Figure 1.4). As a major consequence, the existential quantifier coincide with dependent sum

$$(\exists x \in A)\varphi(x) :\equiv (\Sigma x \in A)\varphi(x)$$

and AC gets validated.

Theorem 1.8.1 will show how to formally implement this specialisation for the kinds of the intensional level, proving that **MLTT** is a compatible extension of **mTT**.

Small	*small type*
	$\downarrow$
Large	*large type*

Figure 1.4: Kinds in **MLTT** relative to a given type universe

Logic-enriched type theory When the types of **MLTT** are used to sort logic, as it happens for example in the logic-enriched type theory of [GA06], the square of the resulting theory (Figure 1.5) shares a closer resemblance with that of **MF**; there, however, logic lacks comprehension, and it is entirely separated from type theory.

	Logic	*Type-theory*
Small	*small prop*	*small type*
	$\downarrow$	$\downarrow$
Large	*prop*	*type*

Figure 1.5: Entities in logic-enriched type theory

Calculus of (Inductive) Constructions In **CIC**, where an impredicative universe of propositions is postulated, the **MF** square is – conversely to **MLTT** – vertically collapsed (see Figure 1.6, where we ignored additional size distinctions given by type universes); consequently, PSA holds, and the existential quantifier is introduced with the usual impredicative encoding.

$$(\exists x \in A)\varphi(x) :\equiv (\forall \psi \in \mathsf{Prop})((\forall x \in A)(\varphi(x) \Rightarrow \psi) \Rightarrow \psi)$$

Analogously to **MLTT**, Theorem 1.8.2 will show how to formally implement this specialisation for the kinds of the intensional level, proving that **CIC** is a compatible extension of **mTT**.

Logic *Set theory*

$$\mathit{prop} \hookrightarrow \mathit{type}$$

Figure 1.6: Kinds in **CIC**

Homotopy Type Theory In **HoTT** (Figure 1.7), both logic and set theory are identified with specific portions of the whole type theory through the characterisation of the types h-levels, which counts how many iterations of the identity types are needed to reach the unit type. The existential quantifier is again defined through dependent sum, this time however it is propositionally truncated to force the right h-level.

$$(\exists x \in A)\varphi(x) :\equiv \|(\Sigma x \in A)\varphi(x)\|$$

In this way, whilst AC does not hold, **HoTT** nevertheless validates AC!.

Due to the peculiar nature of **HoTT**, the specialisation of kinds described above has been formally performed in [CM22] both for the intensional level **mTT**, and for the extensional one **emTT**, showing that they are both compatible with **HoTT**.

Internal language of toposes The internal language $\mathcal{T}_{\mathbf{Topos}}$ formulated as in [Mai09], collapses the square vertically, since it is impredicative, and characterise propositions as the so-called *mono-types*, that is types with at most one element (Figure 1.8); in this way, it validates both AC! and PSA, and existential statements can be defined both using the impredicative encoding as shown for **CIC**, or, as shown in [Mai05], as quotients of dependent sums by the trivial equivalence relation.

$$(\exists x \in A)\varphi(x) :\equiv ((\Sigma x \in A)\varphi(x))/\top$$

$$\begin{array}{lccc} & \textit{Logic} & & \textit{Set theory} \\ \textit{Small} & \textit{small h-prop} & \hookrightarrow & \textit{small h-set} \\ & \downarrow & & \downarrow \\ \textit{Large} & \textit{h-prop} & \hookrightarrow & \textit{h-set} \end{array}$$

Figure 1.7: Kinds in the set-theoretic fragment of **HoTT** relative to a given type universe

In Theorem 1.8.3 we will show how to formally implement this specialisation for the kinds of the extensional level, proving that $\mathcal{T}_{\mathbf{Topos}}$ is a compatible extension of **emTT**.

$$\begin{array}{ccc} \textit{Logic} & & \textit{Set theory} \\ \textit{mono-type} & \hookrightarrow & \textit{type} \end{array}$$

Figure 1.8: Kinds in $\mathcal{T}_{\mathbf{Topos}}$

Constructive set theory The theory **CZF** not only distinguishes between sets and classes, as the classical **ZF** does; it also considers a subclass of formulas, called Δ_0 or *restricted* formulas, again as those in which quantification is performed only over sets; in particular, its axiom of *predicative separation*, allows to create new sets by comprehensions using only Δ_0-formulas. Moreover, by identifying a formula φ with the subsingleton $\{x \mid x = \emptyset \wedge \varphi\}$ it generates by comprehension, one obtains the square in Figure 1.9. The similarity to that of **MF** is evident; and an interpretation of **MF** into **CZF** is easily achieved by specialising each vertex of the square following the standard set-theoretic interpretation of type theory (see [MS22] for a detailed account).

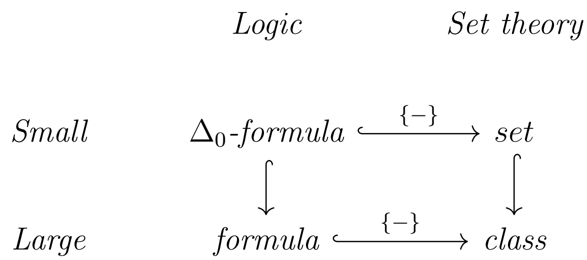


Figure 1.9: Entities in **CZF**

1.6 The formal calculi

In this section, we present the formal calculi **mTT** and **emTT**, formulated as dependent type theories à la Martin-Löf [NPS90], and accounting for the intensional and the extensional level of **MF**, respectively.

Preliminary, we recall that, in dependent type theories, where the definitions of language and derivability are intertwined, the expressions of the calculi have to be introduced first with a so-called *raw-* or *pre-syntax* (see [Str93]). This subtlety is often tacitly overlooked, since, for the sole purpose of presenting a calculus, syntax can be introduced in fieri; however, when formally dealing with operations on syntax such as interpretations and translations, we must resort to it.

We recall that there are ways of presenting type theories which avoid referring to the pre-syntax at all; the syntax of those presentations is usually called *intrinsic* or *typed* (as opposed to the *extrinsic (pre-)syntax* mentioned above) since it coincides at the same time with the derivable judgements of the theory (see [AK16]). The intrinsic approach usually relies on the semantic notion of Dybjer's Categories with Families [Dyb96], by using explicit substitution and making judgemental equality coincide with the equality of the meta-theory. Not to derail from the original presentation of the theory in [Mai09], we followed the extrinsic approach.

Pre-syntax The pre-syntax of both **emTT** and **mTT** consists of pre-contexts, pre-types, pre-propositions, and pre-terms. It is assumed fully annotated, in the sense that each (pre-)term has all the information needed to infer a putative (pre-)type it belongs to. Although for readability we will leave a lot of annotations implicit, in some cases they will be displayed explicitly e.g. to correctly define interpretations.

Substitution of a variable x for a pre-term t in an expression e will be denoted with square brackets $e[t/x]$. Moreover, if we have a list of pre-terms $\gamma \equiv t_1, \ldots, t_n$, we denote with $e[\gamma]$ the successive substitutions $e[t_1/x_1] \cdots [t_n/x_n]$.

Notions of equality When dealing with the syntax of a dependent type theory, we need to carefully distinguish three different notions of equality:

- *meta-equality* between entities in our meta-theory $e_1 \equiv e_2$; it will refer often to equality between expressions of the pre-syntax, and in this case it is called *syntactic equality*; it will be also used to compare the semantical results of interpetations;

- *judgemental equality* between types $A = B$ and terms $a = b \in A$, with the latter sometimes shortened as $a = b$ when the common type is clear from the context;

- the *propositional equality* type between two terms of a given type, which is written $\mathsf{Id}(A, a, b)$ in the intensional level **mTT**, and $\mathsf{Eq}(A, a, b)$ in the extensional one **emTT**; the former will be often abbreviated as $a =_A b$.

Notation We will throughout use the following standard type-theoretic shorthands: we denote with $\rightarrow$ *(resp.* $\times$*)* a non-dependent Π-type *(resp. a non-dependent* Σ*-type)*, and we will reserve the arrow symbol $\Rightarrow$ to denote the implication connective; we will often write $f(a)$ as a shorthand for $\mathsf{Ap}(f, a)$; we define negation, the true constant, and logical equivalence as follows: $\neg\varphi :\equiv \varphi \Rightarrow \bot$, $\top :\equiv \neg\bot$, $\varphi \Leftrightarrow \psi :\equiv \varphi \Rightarrow \psi \wedge \psi \Rightarrow \varphi$.

Form of judgements Both **emTT** and **mTT** have the following forms of dependent judgements. First, we have the auxiliary judgement stating that a telescopic list of variable assumptions is a well-formed context

$$\Gamma\ ctx \quad \text{with } \Gamma \equiv x_1 \in A_1, \ldots, x_n \in A_n(x_1, \ldots, x_{n-1})$$

then, we have typehood judgements, asserting that an expression in context is a type of a certain kind

$$\varphi\ prop_s\ [\Gamma] \qquad \varphi\ prop\ [\Gamma] \qquad A\ set\ [\Gamma] \qquad A\ col\ [\Gamma]$$

respectively, small propositions, propositions, sets, and collections; analogously, there are type equality judgements, asserting that two expressions are equal as types of a certain kind.

$$\varphi = \psi\ \mathit{prop}_s\ [\Gamma] \qquad \varphi = \psi\ \mathit{prop}\ [\Gamma] \qquad A = B\ \mathit{set}\ [\Gamma] \qquad A = B\ \mathit{col}\ [\Gamma]$$

Finally, we have typing and term equality judgements.

$$a \in A\ [\Gamma] \qquad a = b \in A\ [\Gamma]$$

It will also be useful to consider the following judgement forms managing telescopic substitutions

$$\gamma \in \Gamma\ [\Delta] \qquad \gamma = \delta \in \Gamma\ [\Delta]$$

where γ and δ are list of pre-terms, and Γ and Δ are pre-contexts. In particular, the judgement $\gamma \in \Gamma\ [\Delta]$ states that $\gamma \equiv t_1, \ldots, t_n$ is a telescopic substitution from the context Δ to the context $\Gamma \equiv x_1 \in A_1, \ldots, x_n \in A_n$, namely that the following primitive judgements hold.

$$t_1 \in A_1\ [\Delta] \quad \cdots \quad t_n \in A_n[t_1/x_1]\cdots[t_{n-1}/x_{n-1}]\ [\Delta]$$

Analogously, the judgement $\gamma = \delta \in \Gamma\ [\Delta]$ states that $\gamma \equiv t_1, \ldots, t_n$ and $\delta \equiv s_1, \ldots, s_n$ are equal telescopic substitution from the context Δ to the context $\Gamma \equiv x_1 \in A_1, \ldots, x_n \in A_n$, namely that the following primitive judgements hold.

$$t_1 = s_1 \in A_1\ [\Delta] \quad \cdots \quad t_n = s_n \in A_n[t_1/x_1]\cdots[t_{n-1}/x_{n-1}]\ [\Delta]$$

As usual, when doing calculations or writing inference rules, the piece of context common to all the judgements involved is omitted. Moreover, we will often employ the placeholder *type* in a judgement of the form $A\ \mathit{type}\ [\Gamma]$ or $A = B\ \mathit{type}\ [\Gamma]$ standing for one of the four kinds prop_s, *prop*, *set* or *col*; always with the same choice if it occurs multiple times in an inference rule or in a sentence.

Finally, we use the entailment symbol $\mathcal{T} \vdash \mathcal{J}$ to express that *the theory* $\mathcal{T}$ *derives the judgement* $\mathcal{J}$.

Common structural rules The following structural rules are shared by both calculi: rules for context formation and variable assumption

$$\mathsf{ax}\ \frac{}{()\ ctx} \qquad \mathsf{F\text{-}ctx}\ \frac{A\ col\ [\Gamma]}{\Gamma, x \in A\ ctx}\ \text{(with } x \text{ fresh variable)}$$

$$\mathsf{var}\ \frac{\Gamma, x \in A, \Delta\ ctx}{x \in A\ [\Gamma, x \in A, \Delta]}$$

where () denotes the empty context; conversion rules

$$\mathsf{conv}\ \frac{a \in A \quad A = B\ type}{a \in B} \qquad \mathsf{eq\text{-}conv}\ \frac{a = b \in A \quad A = B\ type}{a = b \in B}$$

rules stating that judgemental equalities of types and terms are equivalence relations

$$\mathsf{type\text{-}ref}\ \frac{A\ type}{A = A\ type} \qquad \mathsf{term\text{-}ref}\ \frac{a \in A}{a = a \in A}$$

$$\mathsf{type\text{-}sym}\ \frac{B = A\ type}{A = B\ type} \qquad \mathsf{term\text{-}sym}\ \frac{a = b \in A}{b = a \in A}$$

$$\mathsf{type\text{-}tra}\ \frac{A = B\ type \quad B = C\ type}{A = C\ type} \qquad \mathsf{term\text{-}tra}\ \frac{a = b \in A \quad b = c \in A}{a = c \in A}$$

and, finally, rules for inclusion between kinds of types

$$\mathsf{prop_s\text{-}into\text{-}prop}\ \frac{\varphi\ prop_s}{\varphi\ prop} \qquad \mathsf{prop_s\text{-}into\text{-}set}\ \frac{\varphi\ prop_s}{\varphi\ set}$$

$$\mathsf{set\text{-}into\text{-}col}\ \frac{A\ set}{A\ col} \qquad \mathsf{prop\text{-}into\text{-}col}\ \frac{\varphi\ prop}{\varphi\ col}$$

1.6.1 Minimal Type Theory mTT

The Minimal Type Theory **mTT** is a fully intensional dependent type theory which can be thought as a common refinement of **MLTT** and **CIC**. It is a version of the first-order fragment (meaning the fragment that includes only one universe) of the former enriched with a primitive kind of propositions, and a predicative version of the latter.

In the following, we present its main mechanism and peculiarities; its full set of rules can be found in Appendix A.1.

Structural rules Compared with most dependent type theories, **mTT** follows an unusual design in not assuming judgemental congruence rules for its types and terms constructors. In particular, **mTT** does not assume the so-called ξ-rule for lambda terms.

$$\frac{t(x) = s(x) \in B(x)\ [x \in A]}{(\lambda x \in A)t(x) = (\lambda x \in A)s(x) \in (\Pi x \in A)B(x)}$$

Without further means, the absence of these rules would break a fundamental property of any type system, namely the preservation of judgemental equality under substitution. To compensate for that, **mTT**, additionally to the common structural rules recalled above, postulates the following structural rules of substitutions.

$$\textsf{type-sub}\ \frac{C\ type\ [\Gamma, x \in A, \Delta] \quad a = b \in A\ [\Gamma]}{C[a/x] = C[b/x]\ type\ [\Gamma, \Delta[a/x]]}$$

$$\textsf{term-sub}\ \frac{c \in C\ [\Gamma, x \in A, \Delta] \quad a = b \in A\ [\Gamma]}{c[a/x] = c[b/x] \in C[a/x]\ [\Gamma, \Delta[a/x]]}$$

Then, the types of **mTT** are constructed as follows.

Propositions Propositional constructors are those of predicate logic with equality; namely, the falsum constant $\bot$, the connectives $\wedge$, $\vee$, $\Rightarrow$, the quantifiers $\forall$, $\exists$, and an equality predicate Id à la Leibniz. Their introduction and elimination rules are a proof-relevant, computable version of the corresponding ones in natural deduction, where the only computation rule assumed is β-equality. In particular, their elimination rules act only towards non-dependent propositions; we report for example the elimination and computation rules of the disjunction.

$$\textsf{E-}\vee\ \frac{\chi\ prop \quad p \in \varphi \vee \psi \quad t(x) \in \chi\ [x \in \varphi] \quad s(x) \in \chi\ [x \in \psi]}{\mathsf{El}_\vee(p, (x).t, (x).s) \in \chi}$$

$$\beta\mathsf{C}_1\textsf{-}\vee\ \frac{\chi\ prop \quad \varphi \vee \psi\ prop \quad p \in \varphi \quad t(x) \in \chi\ [x \in \varphi] \quad s(x) \in \chi\ [x \in \psi]}{\mathsf{El}_\vee(\mathsf{inl}_\vee(p), (x).t, (x).s) = t(p) \in \chi}$$

$$\beta\mathsf{C}_2\textsf{-}\vee\ \frac{\chi\ prop \quad \varphi \vee \psi\ prop \quad q \in \psi \quad t(x) \in \chi\ [x \in \varphi] \quad s(x) \in \chi\ [x \in \psi]}{\mathsf{El}_\vee(\mathsf{inr}_\vee(q), (x).t, (x).s) = s(q) \in \chi}$$

The same goes for the intensional equality predicate, which, contrary to Martin-Löf's identity type, eliminates towards propositions not depending on a proof-term.

$$\textsf{E-Id}\ \frac{R(x,y)\ prop\ [x \in A, y \in A] \quad p \in \textsf{Id}(A,a,b) \quad t(x) \in R(x,x)\ [x \in A]}{\textsf{El}_{\textsf{Id}}(p,(x).t) \in R(a,b)}$$

Intuitively, the rule above is saying that $\textsf{Id}(A,x,y)$ is the smallest reflexive relation on A. On the other hand, consider the following alternative elimination rule which, acting towards predicates instead of relations, directly displays the Leibniz characterisation of propositional equality.

$$\textsf{E-Id}'\ \frac{P(x)\ prop\ [x \in A] \quad p \in \textsf{Id}(A,a,b) \quad q \in P(a)}{\textsf{El}_{\textsf{Id}}(p,q) \in P(b)}$$

It is easy to check that the two elimination rules are mutually encodable: one can encode E-Id through E-Id′ by fixing a term of the relation $P(x) :\equiv R(a,x)$; in the other direction, one can use E-Id to encode E-Id′ by exploiting the implication connective $R(x,y) :\equiv P(x) \Rightarrow P(y)$.

Small propositions Small propositions are just those propositions in which quantifications and propositional equalities have been applied only over sets. Formally, they are generated using the following formation rules.

$$\textsf{F}_\textsf{s}\text{-}\bot\ \frac{}{\bot\ prop_s} \qquad \textsf{F}_\textsf{s}\text{-Id}\ \frac{A\ set \quad a \in A \quad b \in A}{\textsf{Id}(A,a,b)\ prop_s}$$

$$\textsf{F}_\textsf{s}\text{-}\wedge\ \frac{\varphi\ prop_s \quad \psi\ prop_s}{\varphi \wedge \psi\ prop_s} \qquad \textsf{F}_\textsf{s}\text{-}\vee\ \frac{\varphi\ prop_s \quad \psi\ prop_s}{\varphi \vee \psi\ prop_s} \qquad \textsf{F}_\textsf{s}\text{-}\Rightarrow\ \frac{\varphi\ prop_s \quad \psi\ prop_s}{\varphi \Rightarrow \psi\ prop_s}$$

$$\textsf{F}_\textsf{s}\text{-}\forall\ \frac{\varphi(x)\ prop_s\ [x \in A] \quad A\ set}{(\forall x \in A)\varphi(x)\ prop_s} \qquad \textsf{F}_\textsf{s}\text{-}\exists\ \frac{\varphi(x)\ prop_s\ [x \in A] \quad A\ set}{(\exists x \in A)\varphi(x)\ prop_s}$$

Sets The sets of **mTT** are formed using the basic type formers of **MLTT** presented with their usual formation, introduction, elimination, and computation rules; namely, we have the empty set N_0, the singleton set N_1, lists List, disjoint sums $+$, dependent sums Σ, and dependent products Π. With the exception of Π-types, the elimination of all the other inductive set constructors is towards dependent collections. It is important to notice that, as it happens for propositions, the only computation rules assumed for those constructors are β-equalities. The η-equalities of set constructors can be derived only propositionally using the elimination principle of the corresponding constructor; as an example, consider the case of dependent sum.

$$\mathsf{El}_\Sigma(c,(x,y).\mathsf{id}(\langle x,y\rangle)) \in \mathsf{Id}(\ (\Sigma x \in A)B(x)\ ,\ c\ ,\ \langle \pi_1(c), \pi_2(c)\rangle\ )$$

Collections Apart from sets and proposition (which are all collections), in **mTT** there is only one basic constructor for collections, namely the universe of small propositions $\mathsf{Prop_s}$. To simplify our meta-mathematical investigations, we consider its version à la Tarski (as opposed to its definition à la Russell in [Mai09]) through the following rules.

$$\textsf{F-Prop}_\mathsf{s}\ \frac{}{\mathsf{Prop_s}\ col} \qquad \textsf{I-Prop}_\mathsf{s}\ \frac{\varphi\ prop_s}{\widehat{\varphi} \in \mathsf{Prop_s}} \qquad \textsf{E-Prop}_\mathsf{s}\ \frac{c \in \mathsf{Prop_s}}{\mathsf{T}(c)\ prop_s}$$

$$\textsf{C-Prop}_\mathsf{s}\ \frac{\varphi\ prop_s}{\mathsf{T}(\widehat{\varphi}) = \varphi\ prop_s} \qquad \eta\textsf{-Prop}_\mathsf{s}\ \frac{c \in \mathsf{Prop_s}}{\widehat{\mathsf{T}(c)} = c \in \mathsf{Prop_s}}$$

$$\textsf{Eq-Prop}_\mathsf{s}\ \frac{\varphi = \psi\ prop_s}{\widehat{\varphi} = \widehat{\psi} \in \mathsf{Prop_s}}$$

Notice that thanks to the structural rule type-sub, also the following rule is derivable.

$$\textsf{Eq-E-Prop}_\mathsf{s}\ \frac{c = d \in \mathsf{Prop_s}}{\mathsf{T}(c) = \mathsf{T}(d)\ prop_s}$$

In this work, we choose a homogeneous coding, contrary to the presentation in [MM15] which instead postulates an ad hoc code for each propositional constructor. The two presentations are essentially equivalent, but the one used here is more suitable to possible extension with new propositional constructors such as those we will consider in Chapter 2 because we won't need to postulate a new introduction term for $\mathsf{Prop_s}$ anytime we introduce a new small propositional constructor. Collections are closed under dependent sum Σ – this is vital to define collections by comprehension with propositions; on the contrary, in general collections, are not closed under the other constructors listed above for sets; and, in particular, they are not closed under dependent products Π. The only exception is that it is possible to form the collection of small propositional operations taking arguments in sets, which can be thought of as a universe of small predicates on that set.

$$\textsf{F-}\Pi\textsf{Prop}_\mathsf{s}\ \frac{A\ set}{A \to \mathsf{Prop_s}\ col}$$

Notice that, contrary to [Mai09], to simplify our meta-mathematical treatment we do not present the above collections through dedicated rules: the arrow $\rightarrow$ refers as usual to the non-dependent version of Π, and the resulting type automatically follows its rules.

1.6.2 Extensional Minimal Type Theory emTT

The Extensional Minimal Type Theory **emTT** is a fully extensional dependent type theory obtained by enriching the first-order fragment of **eMLTT** in [Mar84] with primitive propositions, a quotient set constructor and a predicative powerset constructor; following the predicative terminology of **MF**, the latter will be referred to as a power-*collection* constructor.

In the following, we present its main mechanism and peculiarities; its full set of rules can be found in Appendix A.2.

Strctural rules Apart from the common structural rules shared with **mTT**, the calculus **emTT** has also the following rules for embedding kind equalities

$$\textsf{eq-prop}_\textsf{s}\textsf{-into-prop}\ \frac{\varphi = \psi\ prop_s}{\varphi = \psi\ prop} \qquad \textsf{eq-prop}_\textsf{s}\textsf{-into-set}\ \frac{\varphi = \psi\ prop_s}{\varphi = \psi\ set}$$

$$\textsf{eq-set-into-col}\ \frac{A = B\ set}{A = B\ col} \qquad \textsf{eq-prop-into-col}\ \frac{\varphi = \psi\ prop}{\varphi = \psi\ col}$$

We remark that the structural rules $\textsf{type-sub}$ and $\textsf{term-sub}$ peculiar of **mTT** will nevertheless be derivable in **emTT**.

Then, types of **emTT** are constructed as follows.

Propositions In **emTT**, proof-irrelevance is forced with the following rule.

$$\textsf{prop-mono}\ \frac{\varphi\ prop \quad p \in \varphi \quad q \in \varphi}{p = q \in \varphi}$$

Moreover, a canonical proof-term $\textsf{true}$ is introduced as a useful shorthand.

$$\textsf{prop-true}\ \frac{\varphi\ prop \quad p \in \varphi}{\textsf{true} \in \varphi}$$

We will often render the judgement $\textsf{true} \in \varphi\ [\Gamma]$ as $\varphi\ true\ [\Gamma]$; moreover, when we say that a certain proposition φ *holds*, *can be derived*, *is satisfied* or similar, we mean that the judgement $\varphi\ true\ [\Gamma]$ is derivable.

Propositional constructors are again those of predicate logic with equality, formalised similarly as in **mTT**. Compare for example the elimination rule of disjunction with the same rule reported above for **mTT**.

$$\mathsf{E}\text{-}\vee\ \frac{\xi\ prop \quad \varphi \vee \psi\ true \quad \xi\ true\ [x \in \varphi] \quad \xi\ true\ [x \in \psi]}{\xi\ true}$$

The most important difference with respect to **mTT** is that **emTT** has an *extensional* propositional equality; it is denoted as $\mathsf{Eq}(A, a, b)$ with $a, b \in A$ and its elimination rule is the so-called equality reflection.

$$\mathsf{E}\text{-}\mathsf{Eq}\ \frac{\mathsf{Eq}(A, a, b)\ true}{a = b \in A}$$

Moreover, notice that, since there is proof-irrelevance, propositional constructors do not need computation rules.

Small propositions Exactly as in **mTT**, small propositions are the ones where equality and quantification are restricted to sets.

Sets The set constructors of **mTT** (namely, the empty set N_0, the singleton set N_1, lists List, disjoint sums $+$, dependent sums Σ, and dependent products Π) are present also in **emTT**, with two main differences regarding their equalities. Firstly, **emTT** assumes η-equality rule for dependent products

$$\eta\mathsf{C}\text{-}\Pi\ \frac{f \in (\Pi x \in A)B(x)}{(\lambda x \in A)\mathsf{Ap}(f, x) = f \in (\Pi x \in A)B(x)}$$

As in **mTT**, the η-equality rules of the other set constructors can be derived; now however, thanks to equality reflection, also the judgemental rule is validated.

Secondly, for each type and term constructor, **emTT** assumes its congruence rule; for example, in the case of dependent product there are the following three rules

$$\mathsf{eq}\text{-}\mathsf{F}\text{-}\Pi\ \frac{\begin{array}{l} A\ set \quad B(x)\ set\ [x \in A] \\ A'\ set \quad B'(x)\ set\ [x \in A'] \\ A = A'\ set \quad B(x) = B'(x)\ set\ [x \in A] \end{array}}{(\Pi x \in A)B(x) = (\Pi x \in A')B'(x)\ set}$$

$$\xi\text{eq-I-}\Pi\ \frac{b(x) = b'(x) \in B(x)\ [x \in A] \quad (\Pi x \in A)B(x)\ col}{(\lambda x \in A)b(x) = (\lambda x \in A)b'(x) \in (\Pi x \in A)B(x)}$$

$$\text{eq-E-}\Pi\ \frac{f = f' \in (\Pi x \in A)B(x) \quad a = a' \in A}{\mathsf{Ap}(f, a) = \mathsf{Ap}(f', a') \in B(a)}$$

where the second one is the so-called ξ-equality rule; recall that the combination of the ξ and η congruence rules for the dependent product allows us to derive the following rule

$$\frac{\mathsf{Ap}(f, x) = \mathsf{Ap}(g, x) \in B(x)\ [x \in A]}{f = g \in (\Pi x \in A)B(x)}$$

which, thanks to equality reflection, is equivalent to funext.

Finally, **emTT** has an additional set constructor that allows to form the quotient of a set by a small equivalence relation (that is, an equivalence relation which is a small proposition); its formation rule reads

$$\text{F-Q}\ \frac{\begin{array}{l} A\ set \quad R(x, y)\ prop_s\ [x \in A, y \in A] \\ R(x, x)\ true\ [x \in A] \\ R(y, x)\ true\ [x, y \in A, p \in R(x, y)] \\ R(x, z)\ true\ [x, y, z \in A, p \in R(x, y), q \in R(y, z)] \end{array}}{A/R\ set}$$

The canonical elements of a quotient set are equivalence classes represented by an explicit element of A.

$$\text{I-Q}\ \frac{A/R\ set \quad a \in A}{[a] \in A/R}$$

The congruence rule for the introduction term of quotients is stronger than the usual pattern followed by other constructors.

$$\text{eq-I-Q}\ \frac{A/R\ set \quad a \in A \quad b \in A \quad R(a, b)\ true}{[a] = [b] \in A/R}$$

(notice that in [Mai09] the above rule is called eq-Q, and a weaker eq-I-Q is also stated, although superfluous). In Proposition 1.7.8, quotients will be shown to be *effective*, in the sense that the following derived rule, inverse to eq-I-Q, is derivable.

$$\text{eff}\ \frac{[a] = [b] \in A/R}{R(a, b)\ true}$$

Finally, the quotient elimination and computation rules state that we can eliminate whenever the operation involved does not depend on the choice of representatives.

$$\mathsf{E\text{-}Q}\ \frac{\begin{array}{l} M(z)\ col\ [z \in A/R] \\ c \in A/R \quad m(x) \in M([x])\ [x \in A] \\ m(x) = m(y) \in M([x])\ [x \in A, y \in A, p \in R(x,y)] \end{array}}{\mathsf{El_Q}(c, (x).m) \in M(c)}$$

$$\beta\mathsf{C\text{-}Q}\ \frac{\begin{array}{l} M(z)\ col\ [z \in A/R] \\ a \in A \quad m(x) \in M([x])\ [x \in A] \\ m(x) = m(y) \in M([x])\ [x \in A, y \in A, p \in R(x,y)] \end{array}}{\mathsf{El_Q}([a], (x).m) = m(a) \in M([a])}$$

Notice that for the premises $m(x) = m(y) \in M([x])$ above to typecheck we need the congruence rule eq-I-Q.

Collections Apart from sets and proposition (which are all collections), in **emTT** there is only one basic constructor for collections, the *power collection of the singleton* $\mathcal{P}(1)$. It is also known as the *collection of small propositions up to equiprovability*; indeed, in the quotient model in [Mai09], the collection $\mathcal{P}(1)$ is interpreted in the intensional level as the setoid $(\mathsf{Prop_s}, \Leftrightarrow)$. The fact that its intended interpretation is that of a quotient is evident also by the following rules, which closely match those of the quotient set constructor.

$$\mathsf{I\text{-}}\mathcal{P}(1)\frac{\varphi\ prop_s}{[\varphi] \in \mathcal{P}(1)} \qquad \mathsf{eq\text{-}I\text{-}}\mathcal{P}(1)\frac{\varphi \Leftrightarrow \psi\ true}{[\varphi] = [\psi] \in \mathcal{P}(1)} \qquad \mathsf{eff\text{-}}\mathcal{P}(1)\frac{[\varphi] = [\psi] \in \mathcal{P}(1)}{\varphi \Leftrightarrow \psi\ true}$$

One peculiarity of $\mathcal{P}(1)$ is that, although it is a collection, its equality is postulated to be a small proposition.

$$\mathsf{F_s\text{-}Eq\text{-}}\mathcal{P}(1)\ \frac{U \in \mathcal{P}(1) \quad V \in \mathcal{P}(1)}{\mathsf{Eq}(\mathcal{P}(1), U, V)\ prop_s}$$

Thanks to that, we have a canonical way to pick a representative from the elements $U \in \mathcal{P}(1)$ thought of as equivalence classes of equiprovable small propositions; we define it through the following shorthand

$$\mathsf{Dc}(U) :\equiv \mathsf{Eq}(\mathcal{P}(1), U, [\top])\ prop_s$$

observing that, thanks to the rules $\mathsf{eq\text{-}I\text{-}}\mathcal{P}(1)$ and $\mathsf{eff\text{-}}\mathcal{P}(1)$ above, it satisfies the following derived rule.

$$\frac{\varphi\ prop_s}{\mathsf{Dc}([\varphi]) \Leftrightarrow \varphi\ true}$$

Finally, we postulate the following η-equality, which will be vital to derive subset extensionality in Subsection 1.7.9.

$$\eta\text{-}\mathcal{P}(1)\frac{U \in \mathcal{P}(1)}{[\mathsf{Dc}(U)] = U \in \mathcal{P}(1)}$$

In **emTT**, collections are closed under dependent sum Σ, and under taking the non-dependent function space of a set A towards the collection $\mathcal{P}(1)$.

$$\mathsf{F\text{-}}\Pi\mathcal{P}(1)\ \frac{A\ set}{A \to \mathcal{P}(1)\ col}$$

This last rule is particularly important, since $A \to \mathcal{P}(1)$ will be interpreted as the power-collection of A, which will be throughout shorthanded as follows.

$$\mathcal{P}(A) :\equiv A \to \mathcal{P}(1)$$

1.7 What is like to do mathematics in it?

The *it* in the title of this section should refer to the extensional level **emTT**, since, as we argued in Section 1.4, *it* is the right level of the Minimalist Foundation for developing mathematics. In the following, we collect some basic results useful both for working within **emTT**, and for reasoning about its meta-mathematics; at the same time, we hope that this will give the reader an opportunity to get acquainted with the peculiarities of the system through hands-on examples.

1.7.1 Different notions of function

Recall that in **emTT**, for two sets A and B, and a relation $R(x,y)\ prop\ [x \in A, y \in B]$ between them, the Axiom of Unique Choice $\mathsf{AC!}_{A,B}$, and therefore the Axiom of Choice $\mathsf{AC}_{A,B}$ do not hold in general [Mai17] (see also Proposition 1.7.7 for another small example).

$$(\forall x \in A)(\exists y \in B)R(x,y) \Rightarrow (\exists f \in A \to B)(\forall x \in A)R(x, f(x)) \quad (\mathsf{AC}_{A,B})$$

$$(\forall x \in A)(\exists! y \in B)R(x,y) \Rightarrow (\exists f \in A \to B)(\forall x \in A)R(x, f(x)) \quad (\mathsf{AC!}_{A,B})$$

One of the most important distinctions enforced by the absence of unique choice is the one between functions as type-theoretic functions, which we will call *operations*, that is terms of the function space $A \to B$, and functions as *functional relations*, that is total and singled-value relations on $A \times B$.

In **emTT**, each operation $f \in A \to B$ induces a functional relation by taking its graph

$$f(x) =_B y \; prop_s \; [x \in A, y \in B]$$

However, due to the failure of $\mathsf{AC!}$, not every functional relation can be turned into an operation. Another key difference is that, given two sets A and B, while operations form the set $A \to B$, most of the time small functional relations form a collection.

$$\mathsf{FunRel}(A,B) :\equiv (\Sigma R \in \mathcal{P}(A \times B))(\forall x \in A)(\exists! y \in B)R(\langle x, y\rangle) \; col$$

Even worse, since there is no universe of arbitrary propositions, arbitrary functional relations do not have a type at all in **emTT**.

Finally, whenever A and B are not necessarily sets, it is useful to consider *functional terms* between them, that is terms $t(x) \in B \; [x \in A]$ of type B defined in the context extended by A. Clearly, operations $f \in A \to B$ can be thought as functional terms $\mathsf{Ap}(f, x) \in B \; [x \in A]$, and, whenever the function space $A \to B$ exists, there is an obvious one-to-one correspondence up to judgemental equality between the two notions coincide.

For more analyses of such pivotal distinction, see [MS05] and Sections 1.1.6, 2.1.3, and 2.2.2 of [Sam24]. Throughout this work, we will give greater consideration to the notion of operation and functional terms; starting with the following definitions.

Definition 1.7.2. Let A and B be two collections, and $t(x) \in B \; [x \in A]$ a functional term between them.

We say that t is *injective* if it satisfies

$$(\forall x, y \in A)(t(x) =_B t(y) \Rightarrow x =_A y)$$

surjective if it satisfies

$$(\forall y \in B)(\exists x \in A)t(x) =_B y$$

and *bijective* if it is both injective and surjective, namely, if it satisfies

$$(\forall y \in B)(\exists! x \in A)t(x) =_B y$$

where we have used the shorthand

$$(\exists! x \in A)\varphi(x) :\equiv (\exists x \in A)\varphi(x) \wedge (\forall x, x' \in A)(\varphi(x) \wedge \varphi(x') \Rightarrow x =_A x')$$

In the latter case we say that the sets A and B are *in bijection*.

Moreover, we say that t is an *isomorphism*, or *invertible*, if there exists another functional term $t^{-1}(y) \in A\ [y \in B]$ such that

$$(\forall x \in A)t^{-1}(t(x)) =_A x \ \wedge \ (\forall y \in B)t(t^{-1}(y)) =_B y$$

in this case, we say that A and B are *isomorphic*.

An operation will be called *injective (resp. surjective, bijective, invertible)* if it is injective (surjective, bijective, invertible) as a functional term.

Finally, if A and B are sets, we define the following small proposition shortened as $A \cong B$ expressing internally that A and B are isomorphic.

$$(\exists f \in A \to B)(\exists f^{-1} \in B \to A)(f^{-1} \circ f =_{A \to B} \mathsf{id}_A \ \wedge \ f \circ f^{-1} =_{B \to A} \mathsf{id}_B)$$

where identities $\mathsf{id}_{(-)}$ and compositions $(-) \circ (-)$ for operations are defined in the usual way.

Notice that for a functional term being invertible implies being bijective, but asking in general for the reverse direction is equivalent to $\mathsf{AC!}$.

The notion of isomorphic collections is vital also in the meta-mathematical study of **MF**; in particular, we will often employ the following definition.

Definition 1.7.3. We say that a collection is *proper* if it is not isomorphic to any set.

As expected examples, we have the following.

Proposition 1.7.4. *The collections* $\mathcal{P}(1)$ *and* $\mathcal{P}(\mathbb{N})$ *are proper.*

Proof. If $\mathcal{P}(1)$ were isomorphic to a set, then $\mathcal{P}(\mathbb{N})$ would be too. However, in that case we could interpret full intuitionistic second-order arithmetic in **emTT**; but this is a contradiction since we know that the proof-theoretic strength of **emTT** is bounded by $\widehat{\mathbf{ID}}_1$ as shown in [IMMS18]. □

As non-examples we can just consider sets, or trivialities such as $\mathcal{P}(\mathbb{N}) \times \mathsf{N}_0$ or $\mathcal{P}(\mathsf{N}_0)$, which cannot be proved to be sets but are nevertheless isomorphic to the empty set N_0 and the singleton set N_1, respectively.

1.7.5 Interaction between logic and type theory

Recall that, in the Minimalist Foundation logic is part of the type theory, and thanks to equality, comprehensions, quotients, and power-collections, they are highly intertwined. We will therefore derive some basic properties of this interaction.

For the next proposition we define the following. For a collection A *col*, let $\mathsf{Inh}(A) :\equiv (\exists x \in A)\top$ *prop* be the proposition stating that A is inhabited; define the set of booleans $\mathsf{Bool} :\equiv \mathsf{N}_1 + \mathsf{N}_1$ *set*, with canonical elements $0_{\mathsf{Bool}} :\equiv \mathsf{inl}(\star)$ and $1_{\mathsf{Bool}} :\equiv \mathsf{inr}(\star)$ (we will often omit their subscripts when it is clear from the context).

Proposition 1.7.6 (Logical equivalences)**.** *The following are true in* **emTT***, where φ and ψ denote two propositions defined in the same context and z denotes a fresh variable.*

$$\begin{aligned}
\varphi &\Leftrightarrow \mathsf{Inh}(\varphi)\\
\bot &\Leftrightarrow \mathsf{Inh}(\mathsf{N}_0)\\
\top &\cong \mathsf{N}_1\\
\varphi \wedge \psi &\Leftrightarrow (\exists z \in \varphi)\psi\\
\varphi \wedge \psi &\text{ is isomorphic to } \varphi \times \psi\\
\varphi \vee \psi &\Leftrightarrow \mathsf{Inh}(\varphi + \psi) \quad \text{(with } \varphi \text{ and } \psi \text{ } prop_s\text{)}\\
\varphi \vee \psi &\Leftrightarrow (\exists b \in \mathsf{Bool})((b =_{\mathsf{Bool}} 0 \Rightarrow \varphi) \wedge (b =_{\mathsf{Bool}} 1 \Rightarrow \psi))\\
(\varphi \Rightarrow \psi) &\cong \varphi \to \psi \quad \text{(with } \varphi \text{ and } \psi \text{ } prop_s\text{)}\\
(\varphi \Rightarrow \psi) &\Leftrightarrow (\forall z \in \varphi)\psi\\
(\varphi \Leftrightarrow \psi) &\Leftrightarrow (\varphi \cong \psi) \quad \text{(with } \varphi \text{ and } \psi \text{ } prop_s\text{)}
\end{aligned}$$

Moreover, for a dependent proposition $\varphi(x)$ prop $[x \in A]$ the following hold.

$$\begin{aligned}
(\exists x \in A)\varphi(x) &\Leftrightarrow \mathsf{Inh}((\Sigma x \in A)\varphi(x))\\
(\forall x \in A)\varphi(x) &\cong (\Pi x \in A)\varphi(x) \quad \text{(with } \varphi \text{ } prop_s \text{ and } A \text{ } set\text{)}
\end{aligned}$$

Proof. The condition of smallness required for φ and ψ ensures well-typing when set constructors $+$, Π, and $\cong$ are used in a statement.

All points are trivially checked. It is just a matter of playing around with the various elimination principles. □

In the proposition above, we could not have hoped for a full isomorphism $\bot \cong \mathsf{N}_0$, as the next proposition shows.

Proposition 1.7.7. *The falsum constant* $\perp$ *and the empty set* N_0 *are not isomorphic.*

Proof. We show that there are no operations from $\perp$ to N_0, that is no terms of type $\perp \to \mathsf{N}_0$ (in the empty context). The proof adapts the interpretation already used in [Smi88] to prove the independence of Peano's fourth axiom from Martin-Löf's type theory without universes. Namely, we interpret each type as a boolean value; in particular, we interpret each non-propositional constructor as in the original interpretation; while all the propositional constructors and the collection $\mathcal{P}(1)$ are interpreted as 1. Since propositions eliminates only towards propositions, it is easy to prove in the same way as in [Smi88] that if A is a closed type such that **emTT** $\vdash a \in A$ for some term a, then A gets interpreted as 1; however, the type $\perp \to \mathsf{N}_0$ is interpreted as $1 \to 0 \equiv 0$. □

It is interesting to notice that, as it happens for Peano's fourth axiom in Martin-Löf's type theory, the additional presence of a type universe with codes for N_0 and N_1 is enough to derive $\perp \cong \mathsf{N}_0$.

The following proposition is a crucial one, since it characterises the equality of type constructors.

Proposition 1.7.8 (Equality of type constructors)**.** *The following equivalences hold in* **emTT** *(where the free variables in the left hand side of each equivalence are implicitly assumed to be in the obvious context):*

1. $x =_{\mathsf{N}_0} y \Leftrightarrow \perp$

2. $x =_{\mathsf{N}_1} y \Leftrightarrow \top$

3. $[a] =_{A/R} [b] \Leftrightarrow R(a,b)$

4. $l =_{\mathsf{List}(A)} l' \Leftrightarrow \begin{cases} \top & \textit{if } l = l' = \epsilon \\ s =_{\mathsf{List}(A)} s' \wedge a =_A a' & \textit{if } l = \mathsf{cons}(s,a) \textit{ and } l' = \mathsf{cons}(s',a') \\ \perp & \textit{otherwise} \end{cases}$

5. $z =_{A+B} z' \Leftrightarrow \begin{cases} x =_A x' & \textit{if } z = \mathsf{inl}(x) \textit{ and } z' = \mathsf{inl}(x') \\ y =_B y' & \textit{if } z = \mathsf{inr}(y) \textit{ and } z' = \mathsf{inr}(y') \\ \perp & \textit{otherwise} \end{cases}$

6. $z =_{(\Sigma x \in A)B(x)} w \Leftrightarrow (\exists p \in \pi_1(z) =_A \pi_1(w))\, \pi_2(z) =_{B(\pi_1(z))} \pi_2(w)$

7. $f =_{(\Pi x \in A)B(x)} g \Leftrightarrow (\forall x \in A) f(x) =_{B(x)} g(x)$

8. $U =_{\mathcal{P}(1)} V \Leftrightarrow (\mathsf{Dc}(U) \Leftrightarrow \mathsf{Dc}(V))$

9. $p =_{\varphi} q \Leftrightarrow \top \quad$ *if* φ *prop*

Proof. 1. Trivial, using the elimination principle of N_0 towards the equivalence itself; in fact, the choice of the falsum constant at the right hand side of the equivalence is purely conventional.

2. This is the easily derivable η-equality for the singleton set.

3. One direction is just the application of the congruence rule for equivalence classes eq-I-Q. The other direction holds because in **emTT** quotients are effective, in the sense that the following rule is derivable.

$$\mathsf{eff}\ \frac{a \in A \quad b \in A \quad [a] = [b] \in A/R}{R(a,b)\ true}$$

The presentation in [Mai09] of **emTT** postulates the above rule, together with effectiveness for the collection $\mathcal{P}(1)$

$$\mathsf{eff}\text{-}\mathcal{P}(1)\ \frac{[\varphi] = [\psi] \in \mathcal{P}(1)}{\varphi \Leftrightarrow \psi\ true}$$

however, we can actually derive the former using the latter. Indeed, suppose to have $[a] = [b] \in A/R$ for some $a, b \in A$; plugging them in the functional term

$$\mathsf{El_Q}(z, (x).[R(a,x)]) \in \mathcal{P}(1)\ [z \in A/R]$$

results in the equality $[R(a,a)] = [R(a,b)] \in \mathcal{P}(1)$. By effectivity of $\mathcal{P}(1)$ we conclude $R(a,a) \Leftrightarrow R(a,b)$, and thus $R(a,b)$ by the reflexivity of R.

4. The proposition defined by cases on the right side can be formally defined by recursion using the elimination principle of lists towards $\mathcal{P}(1)$, together with coding and decoding functions (here the smallness of propositions is vital).

$$\begin{aligned}&\mathsf{Dc}(\mathsf{El_{List}}(l\\ &\qquad\qquad , \mathsf{El_{List}}(l', [\top], [\bot])\\ &\qquad\qquad , (s,a).\mathsf{El_{List}}(l', [\bot], (s',a').[s =_{\mathsf{List}(A)} s' \wedge a =_A a'])))\end{aligned}$$

The equivalence can then be reduced by induction (again with list elimination) to the following ones:

- $\epsilon =_{\mathsf{List}(A)} \epsilon \Leftrightarrow \top$
- $s =_{\mathsf{List}(A)} s' \wedge a =_A a' \Leftrightarrow \mathsf{cons}(s, a) =_{\mathsf{List}(A)} \mathsf{cons}(s', a')$
- $\epsilon =_{\mathsf{List}(A)} \mathsf{cons}(s, a) \Leftrightarrow \bot$

The first two are trivially checked using congruence rules. For the third one, we can prove, using the same trick of point 3, that list term constructors are disjoint, meaning that the following rule is derivable.

$$\frac{l \in \mathsf{List}(A) \quad a \in A \quad \mathsf{cons}(l, a) = \epsilon \in \mathsf{List}(A)}{\bot \ \mathit{true}}$$

5. Analogous to point 4. In particular, the cases on the right side can be formally defined using the elimination of sum.

$$\begin{aligned}&\mathsf{Dc}(\mathsf{El}_+(z\\ &\qquad , (x).\mathsf{El}_+(z, (x').[x =_A x'], [\bot])\\ &\qquad , (y).\mathsf{El}_+(z, [\bot], (y').[y =_B y'])))\end{aligned}$$

Moreover, we can also prove that the term constructors of the disjoint sum are, indeed, disjoint.

$$\frac{a \in A \quad b \in B \quad \mathsf{inl}(a) = \mathsf{inr}(b) \in A + B}{\bot \ \mathit{true}}$$

6. The equivalence is easily proven using congruence rules, and the derivable η-equality for dependent sums. However, it is important to notice that the right hand side of the equivalence could not have been written as the conjuction $\pi_1(z) =_A \pi_1(w) \wedge \pi_2(z) =_{B(\pi_1(z))} \pi_2(w)$, which is ill-formed because the judgement $\pi_2(w) \in B(\pi_1(z))$ cannot be derived without assuming $\pi_1(z) =_A \pi_1(w)$ first.

7. This is function extensionality, which is proven as usual with equality reflection, and the rules of ξ-equality and η-equality for dependent function spaces.

8. Easily proven using the congruence rule and the η-equality of $\mathcal{P}(1)$.

9. Trivial, by the rule prop-mono.

□

1.7.9 Basic set theory

We are now in the position to show through examples how the development of some basic set theory can be performed smoothly in **emTT**.

Local (sub)set theory As it happens in the internal language of a topos, we can develop *local (sub)set theory* relative to a given set A. We use the shorthand

$$a \,\varepsilon\, V :\equiv \mathsf{Dc}(\mathsf{Ap}(V, a))\ prop_s$$

to denote the (propositional) relation of membership between terms $a \in A$ and subsets $V \in \mathcal{P}(A) \equiv A \to \mathcal{P}(1)$. As an immediate corollary of Proposition 1.7.8, our setting validates subset extensionality

$$U =_{\mathcal{P}(A)} V \Leftrightarrow (\forall x \in A)(x \,\varepsilon\, U \Leftrightarrow y \,\varepsilon\, U)$$

Moreover, we can recover the usual set-builder notation from axiomatic set theory by defining the subset obtained by comprehension with a predicate $\varphi(x)\ prop_s\ [x \in A]$ as follows

$$\{x \in A \,|\, \varphi(x)\} :\equiv (\lambda x \in A)[\varphi(x)] \in \mathcal{P}(A)$$

Notice that it satisfies, for each $a \in A$, the usual comprehension axiom

$$a \,\varepsilon\, \{x \in A \,|\, \varphi(x)\} \Leftrightarrow \varphi(a)$$

We will then use the following set-theoretic shorthands.

$$\begin{aligned}
\emptyset &:\equiv \{x \in A \,|\, \bot\} \\
\{a_1, \ldots, a_n\} &:\equiv \{x \in A \,|\, x =_A a_1 \vee \cdots \vee x =_A a_n\} \\
A &:\equiv \{x \in A \,|\, \top\} \\
U \cap V &:\equiv \{x \in A \,|\, x \,\varepsilon\, U \wedge x \,\varepsilon\, V\} \\
U \cup V &:\equiv \{x \in A \,|\, x \,\varepsilon\, U \vee x \,\varepsilon\, V\} \\
U^{\complement} &:\equiv \{x \in A \,|\, \neg x \,\varepsilon\, U\}
\end{aligned}$$

Finally, given $V \in \mathcal{P}(A)$, and $\varphi(x)\ prop\ [x \in A]$ we define

$$\begin{aligned}
(\forall x \,\varepsilon\, V)\varphi(x) &:\equiv (\forall x \in A)(x \,\varepsilon\, V \Rightarrow \varphi(x))\ prop \\
(\exists x \,\varepsilon\, V)\varphi(x) &:\equiv (\exists x \in A)(x \,\varepsilon\, V \ \wedge\ \varphi(x))\ prop
\end{aligned}$$

Notice that if φ is a small proposition, then also the above are.

Number sets The natural numbers are coded as lists over the singleton set $\mathbb{N} :\equiv \mathsf{List}(\mathsf{N}_1)$. The number zero, the successor operation, and the recursion elimination principles are coded as usual.

$$\begin{aligned} \mathsf{zero} &:\equiv \epsilon \\ \mathsf{succ}(n) &:\equiv \mathsf{cons}(n, \star) \\ \mathsf{El}_{\mathbb{N}}(n, b, (x, y).r(x, y)) &:\equiv \mathsf{El}_{\mathsf{List}}(n, b, (x, y, z).r(x, z)) \end{aligned}$$

Notice that, as an instance of list disjointness (see Proposition 1.7.8), the fourth Peano axiom $\neg\,\mathsf{succ}(\mathsf{zero}) =_{\mathbb{N}} \mathsf{zero}$ is valid.

The set of integers is defined as a quotient $\mathbb{Z} :\equiv (\mathbb{N} \times \mathbb{N})/{\sim_{\mathbb{Z}}}$, where $\langle a, b\rangle \sim_{\mathbb{Z}} \langle c, d\rangle :\equiv a + d =_{\mathbb{N}} b + c$.

Analogously, the set of rational numbers is defined as $\mathbb{Q} :\equiv (\mathbb{Z} \times \mathbb{Z}^*)/{\sim_{\mathbb{Q}}}$, where $\mathbb{Z}^* :\equiv (\Sigma z \in \mathbb{Z})\,\neg z =_{\mathbb{Z}} 0_{\mathbb{Z}}$ is the set of non-zero integers defined by comprehension, and $\langle a, b, p\rangle \sim_{\mathbb{Q}} \langle c, d, q\rangle :\equiv a \cdot d =_{\mathbb{Z}} b \cdot c$.

Observe that all the number sets $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{Q}$ defined until now are indeed sets in the formal sense of **emTT**. Regarding real numbers, they can be defined à la Cauchy $\mathbb{R}_c$, using sequences of rational numbers, or à la Dedekind $\mathbb{R}_d$, using his cuts. The former is the method pursued by Bishop in [MS23b] for his constructive real numbers, which will lead to an effective notion of real number, and the collection of all of them will form a set; the latter is a generalisation of the former closer to the spirit of the continuum proposed by Weyl, where real numbers form a proper collection (see the discussion at Subsection 1.2.3).

Real numbers à la Cauchy are defined using effective sequences or rationals, that is operations in $\mathbb{N} \to \mathbb{Q}$ as follow.

$$\mathbb{R}_c :\equiv ((\Sigma f \in \mathbb{N} \to \mathbb{Q})\mathsf{Reg}(f))/{\sim_{\mathsf{Reg}}}$$

where $\mathsf{Reg}(f)$ is the predicate asserting that a sequence $f \in \mathbb{N} \to \mathbb{Q}$ is regular

$$(\forall n, m \in \mathbb{N}^+)|f(n) - f(m)| \leq_{\mathbb{Q}} n^{-1} + m^{-1}$$

while $f \sim_{\mathsf{Reg}} g$ is the equivalence relation

$$(\forall n \in \mathbb{N}^+)|f(n) - g(n)| \leq_{\mathbb{Q}} n^{-1} + n^{-1}$$

Real numbers à la Dedekind are defined using the collection of subsets of rational numbers $\mathcal{P}(\mathbb{Q})$. For simplicity we report their definition using lower

Dedekind cuts.

$$\begin{aligned}\mathbb{R}_d :\equiv (\Sigma A \in \mathcal{P}(\mathbb{Q}))((\exists q \in \mathbb{Q}) q \,\varepsilon\, A \\ \wedge (\exists q \in \mathbb{Q}) \neg q \,\varepsilon\, A \\ \wedge (\forall q \,\varepsilon\, A)(\forall r \in \mathbb{Q})(r < q \Rightarrow r \,\varepsilon\, A) \\ \wedge (\forall q \,\varepsilon\, A)(\exists r \,\varepsilon\, A) q < r)\end{aligned}$$

In this way, real numbers form a collection, which, in Theorem 4.3.2, we will prove proper.

Finally, we can define an injective functional relation from Cauchy reals $\mathbb{R}_c$ to Dedekind reals $\mathbb{R}_d$ by associating to each regular sequence $f \in \mathbb{N} \to \mathbb{Q}$ the following Dedekind cut.

$$\{q \in \mathbb{Q} \,|\, (\exists m \in \mathbb{N})(\forall n \in \mathbb{N})(n > m \Rightarrow q < f(n))\}$$

Image, inverse image, coimage Given a function $f \in A \to B$ between sets, we define its *image* as follows.

$$\begin{aligned}&\mathsf{Im} f \; \mathit{set} \\ &\mathsf{Im} f :\equiv (\Sigma y \in B)(\exists x \in A) f(x) =_B y\end{aligned}$$

Its elements will be of the form $\langle b, \mathsf{true}\rangle$ for some $b \in B$, so they do not contain any hint to what a possible $a \in A$ in its inverse image could be.

Analogously, the *inverse image* on an element of B is defined as the following dependent set.

$$\begin{aligned}&f^{-1}(y) \; \mathit{set} \; [y \in B] \\ &f^{-1}(y) :\equiv (\Sigma x \in A) f(x) =_B y\end{aligned}$$

Finally, the *coimage* $\mathsf{Coim} f$ of f is defined as the quotient $A/\mathsf{Ker} f$, where the kernel $\mathsf{Ker} f$ is the equivalence relation $f(x) =_B f(y)$. Moreover, notice that we always have a canonical operation

$$\begin{aligned}&u_f \in \mathsf{Coim} f \to \mathsf{Im} f \\ &u_f([x]) :\equiv \langle f(x), \mathsf{true}\rangle\end{aligned}$$

formally defined as $(\lambda z \in \mathsf{Coim} f)\mathsf{El}_{\mathsf{Q}}(z, (x).\langle f(x), \mathsf{true}\rangle)$. It can be easily shown that u_f is bijective, but in general it is not possible to write an inverse function, since that would require to explicitly know, for each $b \in B$, an element in its inverse image.

Relation closures Recall that the *reflexive (resp. symmetric, transitive, equivalence)* closure of a relation $R(x, y)$ *prop* $[x, y \in A]$ is the smallest reflexive (resp. symmetric, transitive, equivalence) relation $S(x, y)$ *prop* $[x, y \in A]$ such that

$$R(x, y) \Rightarrow S(x, y) \; true \; [x, y \in A]$$

The reflexive and symmetric closures can always be constructed as $R(x, y) \vee x =_A y$ and $R(x, y) \vee R(y, x)$, respectively. Moreover, if A is a set, we can construct the *transitive closure* R^+ of R by stating that, for two elements $a, b \in A$, the relation $R^+(a, b)$ holds whenever there exist $x_0, \ldots, x_n \in A$ such that $x_0 =_A a$, $x_n =_A b$, and $R(x_i, x_{i+1})$ for each $i = 0, \ldots, n-1$. This construction can be formalised using the list constructor, and this is why we need to require A to be a set.

By successively applying all the above closure constructions we obtain the equivalence closure of a relation. Notice that, whenever the relation R is small, then also its equivalence closure is.

1.7.10 A categorical account

The question *what is like doing mathematics in a certain foundation?* can be phrased formally with the language of category theory. In particular, we can ask what the category of sets and functions **Set** looks like assuming such foundation. In the case at present, the objects of **Set** are the (closed) sets of **emTT**, considered up to judgemental equality; an arrow $t(x) : A \to B$ from an object A to an object B is a functional term $t(x) \in B \; [x \in A]$, again considered up to judgemental equality.

The following results elaborate on the categorical ones of [Mai09] (in particular Theorem 4.20 thereof), [MPR23] and [Mai05] in the case of **emTT**-sets.

Firstly, we notice that, as expected, there is a tight link between type-theoretical properties of operations and categorical properties of arrows.

Definition 1.7.11. A *comprehension map* in **Set** is an arrow isomorphic to one of the form

$$\pi_1(z) : (\Sigma x \in A)\varphi(x) \to A$$

for a predicate $\varphi(x)$ $prop_s$ $[x \in A]$.

A *quotient map* in **Set** is an arrow isomorphic to one of the form

$$[x] : A \to A/R$$

for an equivalence relation $R(x,y)$ $prop_s$ $[x,y \in A]$.

Proposition 1.7.12. *Let $f : A \to B$ be an arrow in* **Set**.

1. *f is an epimorphism if and only if it is a surjective operation;*
2. *f is a regular epimorphism if and only if it is a quotient map;*
3. *f is an monomorphism if and only if it is an injective operation;*
4. *f is a strong monomorphism if and only if it is a comprehension map.*
5. *f is an isomorphism (in the sense of category theory) if and only if it is an isomorphism as an operation.*

Proof. 1. Assume f is an epimorphism. Consider the relation R_f on the set $B+B$ obtained as the reflexive and symmetric closure of the following proposition.

$$(\exists x \in A)(z =_{B+B} \mathsf{inl}(f(x)) \wedge w =_{B+B} \mathsf{inr}(f(x))) \quad \text{with } z, w \in B+B$$

It is easy to check that R_f is also transitive, and thus an equivalence relation. It is immediate to see that the following diagram commutes.

$$A \xrightarrow{f(x)} B \underset{[\mathsf{inr}(y)]}{\overset{[\mathsf{inl}(y)]}{\rightrightarrows}} (B+B)/R_f$$

Since by hypothesis f is an epimorphism, we deduce

$$(\forall y \in B)[\mathsf{inl}(y)] =_{(B+B)/R_f} [\mathsf{inr}(y)]$$

which is equivalent to $(\forall y \in B)(\exists x \in A)y =_B f(x)$, which precisely expresses that f is surjective.

Conversely, assume $(\forall y \in B)(\exists x \in A)y =_B f(x)$, and suppose to have a commutative diagram of the following form.

$$A \xrightarrow{f(x)} B \underset{h(x)}{\overset{g(x)}{\rightrightarrows}} C$$

Commutativity of the above diagrams reads $(\forall x \in A)f(t(x)) =_B g(t(x))$; with the assumption of surjectivity, we can derive $(\forall y \in B)f(y) =_C g(y)$, which amounts to the equality of the parallel arrows.

2. The fact that each coequaliser is isomorphic to a quotient map is implied by the construction of coequaliser in **Set** shown next in the proof of Theorem 1.7.17. Conversely, notice that each quotient map $[-] : A \to A/R$ is the coequaliser of the pair $\pi_1, \pi_1 \circ \pi_2 : (\Sigma x \in A)(\Sigma y \in A)R(x, y) \to A$.

3. Assume f is a monomorphism, and consider the following diagram.

$$(\Sigma x \in A)(\Sigma y \in A)f(x) =_B f(y) \underset{\pi_1 \circ \pi_2}{\overset{\pi_1}{\rightrightarrows}} A \xrightarrow{f(x)} B$$

   Clearly, it is commutative; since f is a monomorphism, the two parallel arrows are equal, which amounts to the statement of injectivity.

   Conversely, assume f is injective and suppose to have a commutative diagram of the following form.

$$C \underset{h(x)}{\overset{g(x)}{\rightrightarrows}} A \xrightarrow{f(x)} B$$

   Commutativity implies $(\forall x \in C)f(g(x)) =_B f(h(x))$, and by injectivity we deduce $(\forall x \in C)f(x) =_A g(x)$, which amounts to the equality of the parallel arrows.

4. The fact that each equaliser is isomorphic to a comprehension map is implied by the construction of equalisers in **Set** shown next in the proof of Theorem 1.7.17. Conversely, notice that each comprehension map $\pi_1 : (\Sigma x \in A)\varphi(x) \to A$ is the equaliser of the following pair of arrows

$$A \underset{[\mathsf{inr}(x)]}{\overset{[\mathsf{inl}(x)]}{\rightrightarrows}} (A + A)/R_\varphi$$

   where R_φ is the equivalence relation on $A + A$ obtained as the reflexive and symmetric closure (which automatically is also transitive) of the following proposition.

$$(\exists x \in A)(\varphi(x) \wedge z =_{A+A} \mathsf{inl}(x) \wedge w =_{A+A} \mathsf{inr}(x)) \quad \text{with } z, w \in A + A$$

5. Trivial, by definition of inverse operation.

□

Corollary 1.7.13. *The category* **Set** *is not balanced.*

Proof. If it were, by the above characterisation bijective operations would coincide with invertible ones, but we know this is not the case since that would coincide with $\mathsf{AC!}$, which we know does not hold by [Mai17]. □

The following definitions are justified in light of Proposition 1.7.12, which identifies the underlying logic of **Set** with that of *regular* monomorphisms as opposed to e.g. topos theory, in which logic is interpreted through monomorphisms.

Definition 1.7.14. Let $\mathcal{C}$ be a finitely complete category with finite coproducts. We say that $\mathcal{C}$ has *regularly disjoint sums* if, for each pair of objects $A, B \in \mathcal{C}$ the following hold:

- the injections $\iota_1 : A \to A+B$ and $\iota_2 : B \to A+B$ are monomorphisms;
- the unique arrow out of the initial object towards the equaliser of the pair $\iota_1 \circ \pi_1, \iota_2 \circ \pi_2 : A \times B \to A + B$ is an epimorphism; pictorially

$$0 \xrightarrow{!_E} E \rightarrowtail A \times B \underset{\iota_2 \circ \pi_2}{\overset{\iota_1 \circ \pi_1}{\rightrightarrows}} A + B$$

Definition 1.7.15. Let $\mathcal{C}$ be a finitely complete category. A *regular congruence* in $\mathcal{C}$ is a congruence which is also a regular monomorphism.

Remark 1.7.16. Observe that, whenever the category $\mathcal{C}$ is balanced and has a strict initial object, Definition 1.7.14 coincides with the usual ones of disjoint coproducts.

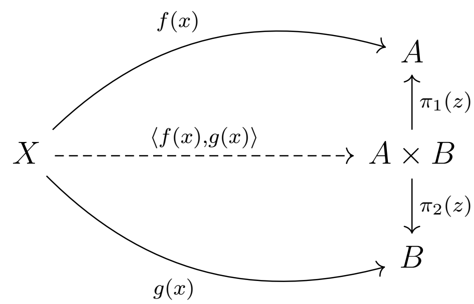
Theorem 1.7.17. *The category* **Set** *is finitely complete, finitely cocomplete, and locally cartesian closed; moreover, it has a natural number object, regularly disjoint sums, and effective quotients of regular congruences.*

Proof. In the following diagrams, a dashed line will denote the unique map induced by an universal property. Unless otherwise stated, uniqueness is proven using the η-equality or the elimination rule of the corresponding constructor.

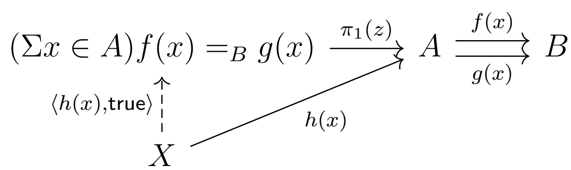
Terminal object

$$X \overset{\star}{\dashrightarrow} \mathsf{N}_1$$

Binary products



equalisers

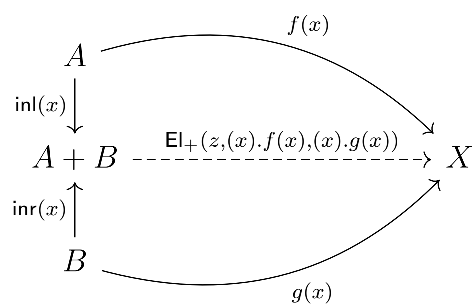


Uniqueness is proven also thanks to proof-irrelevance i.e. the rule prop-mono.

Initial object

$$\mathsf{N}_0 \xrightarrow{\mathsf{El}_{\mathsf{N}_0}(z)} X$$

Regular disjoint sums



Sum disjointness is easily proved using its type-theoretic counterpart of the same name.

Effective quotients of regular congruences

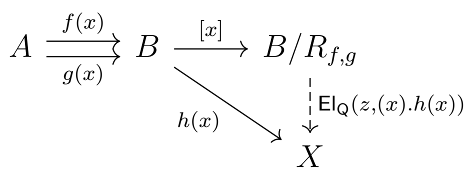
Thanks to point 2 of Proposition 1.7.12, we know that each regular congruence in **Set** is (isomorphic to one) of the following form

$$(\Sigma x \in A)(\Sigma y \in A)R(x,y) \overset{\pi_1}{\underset{\pi_1 \circ \pi_2}{\rightrightarrows}} A$$

where $R(x,y)$ $prop_s$ $[x \in A, y \in A]$ is an equivalence relation (in the type-theoretic sense).

It is easy to see that the quotient map $[x] : A \to A/R$ is the coequaliser of the above diagram, and that $(\Sigma x \in A)(\Sigma y \in A)R(x,y)$ is its kernel pair, using in particular the type-theoretic rule eff of effectivity for the quotients.

Coequalisers



where $R_{f,g}$ is defined as the equivalence closure of the following proposition.

$$(\exists z \in A)(f(z) =_B x \wedge g(z) =_B y) \; prop \; [x \in B, y \in B]$$

To show that the arrow induced by the universal property is well-defined we need to prove $h(x) =_X h(y)$ for each $x, y \in B$ such that $R_{f,g}(x,y)$; notice that $h(x) =_X h(y)$ is an equivalence relation on B; moreover, thanks to the fact that $h(f(x)) =_X h(g(x))$ for any $x \in A$, one has

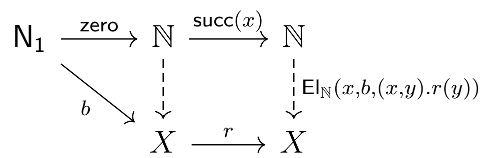
$$(\exists z \in A)(f(z) =_B x \wedge g(z) =_B y) \Rightarrow h(x) =_X h(y)$$

By definition of equivalence closure we conclude that $R_{f,g}(x,y)$ implies $h(x) =_X h(y)$.

Right adjoints to pullback functors Given an arrow $f : A \to B$, the right adjoint $\Pi_f : \mathbf{Set}/A \to \mathbf{Set}/B$ to the pullback functor f^* is defined as in [See84; Law69]; the definition on objects reads

$$\Pi_f(X \overset{g}{\to} A) :\equiv \pi_1(z) : (\Sigma y \in B)(\Pi x \in f^{-1}(y))g^{-1}(\pi_1(x)) \to B$$

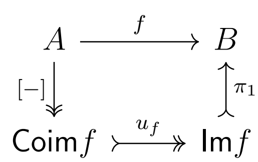
Natural number object



□

Recall that each finitely complete, finitely cocomplete category enjoys two orthogonal factorisation systems given by the pairs of morphism classes (Regular Epimorphisms, Monomorphisms) and (Epimorphisms, Regular Monomorphisms). Moreover, they form a *double factorization system* in the sense of [PATW02], meaning that each arrow factors in an essentially unique way as a regular epimorphism, followed by a morphism that is both an epimorphism and a monomorphism, followed by a regular monomorphism.

In the case of **Set**, each arrow $f : A \to B$ factors uniquely as follows



where $\mathsf{Coim}f$, u_f, and $\mathsf{Im}f$ are defined as in Subsection 1.7.9.

Remark 1.7.18. As already pointed out in Remark 4.41 of [Mai09], the category Q(**mTT**) (Definition 4.1 of [Mai09]) representing the quotient model in which **emTT** is interpreted does not have the latter as its internal language. In categorical terms, we can say that the category **Set** studied in this section is not equivalent to Q(**mTT**). Notably, **Set** does not seem to have enough projectives, whereas Q(**mTT**) does.

1.8 Compatibility results

In this section, we collect the results inspired by the ideas exposed in [Mai19; Mai20] which formally justify the adjective *minimal* in the names of both calculi **mTT** and **emTT**, as well as the adjective *minimalist* in that of **MF**.

For brevity and to help intuition and promote conceptual significance, in the following statements, we refer to the notions of *compatible theory* and *compatible extension* discussed in Section 1.3, and we will implicitly think of them as relative to the intended interpretations of $\mathbf{HA}^{\omega}$ into the theories in question. As mentioned there, those are not precise definitions but more blueprints which can be followed to obtain, in each specific case, a fully formal statement. Example of those unfolded statement can be found for example in [CM22] and [MS22].

Finally, recall that the idea in all the following proofs is always to define an interpretation formalising the specialisation of the entities distinction described for each theory in Section 1.5.

In the following statement, let $\mathbf{ML}_1$ denote the version of **MLTT** in [NPS90] with type constructors $\mathsf{N_0}$, $\mathsf{N_1}$, $+$, List, Σ, Π, Martin-Löf's identity type Id, and one universe of small types U_0 closed under all the other constructors.

Theorem 1.8.1. $\mathbf{ML}_1$ *is a compatible extension of* **mTT** *up to equivalence of theories.*

Proof. Consider the extension of **mTT** obtained by first enforcing the proposition-as-type paradigm through the rules

$$\frac{A\ set}{A\ prop_s} \qquad \frac{A\ col}{A\ prop}$$

and then by adding all the rules of $\mathbf{ML}_1$ that are not already present in **mTT**, namely: congruence rules for type and term constructors; rules for upgrading the Leibniz identity type to Martin-Löf's identity type; and rules closing collections over all type constructors.

Up to renaming, the theory obtained in this way coincide with $\mathbf{ML}_1$: it is enough to rename in the pre-syntax logical symbols under the Curry-Howard isomorphism, and the universe of small propositions $\mathsf{Prop_s}$ as the universe of small types U_0.

Finally, checking compatibility is straightforward, since Martin-Löf's type theory natively interprets logic under the proposition-as-type paradigm. □

In the following statement, let $\mathbf{CC}_{\mathsf{ML}}$ denote the Calculus of Constructions without universes of types (apart from the impredicative universe of propositions) defined in [CH88], extended with rules for the inductive type constructors $\mathsf{N_0}$, $\mathsf{N_1}$, $+$, List, and Σ from the first-order fragment of **MLTT** (notice that the resulting theory is a rather small fragment of the Calculus of Inductive Constructions **CIC** in [CP90]).

Theorem 1.8.2. $\mathbf{CC}_{\mathsf{ML}}$*, and thus* **CIC***, are compatible extensions of* **mTT** *up to equivalence of theories.*

Proof. Consider the impredicative extension of **mTT** obtained by extension with the congruence rules for types and terms and with the following resizing rules collapsing the predicative distinction between effective and open-ended types.

$$\textsf{col-into-set}\ \frac{A\ col}{A\ set} \qquad \textsf{prop-into-prop}_{\mathsf{s}}\ \frac{\varphi\ prop}{\varphi\ prop_s}$$

As consequences of the distinction between sets and collections, as well as that between propositions and small propositions, disappearing we have the following: the universe of small proposition $\mathsf{Prop_s}$ can be interpreted as the impredicative universe of (all) propositions Prop; all types are closed under all set constructors, as in $\mathbf{CC}_{\mathsf{ML}}$ (in particular, the universal quantifier $\forall$ and the dependent function space Π become just two different names for the only Π constructor of $\mathbf{CC}_{\mathsf{ML}}$). Thus, the only calculations to be made to check that the theory obtained is equivalent to $\mathbf{CC}_{\mathsf{ML}}$ are those to verify that propositional constructors of **mTT** are encodable in $\mathbf{CC}_{\mathsf{ML}}$, and this is easily checked using their standard impredicative encodings; in particular, it works for propositional equality because in **mTT** is defined à la Leibniz.

Compatibility is straightforward since the way in which the Calculus of Constructions interprets $\mathbf{HA}^{\omega}$, which does not involve higher order entities, is identical to that of **mTT**. □

In the following statement, let $\mathcal{T}_{\mathbf{ArithTopos}}$ be the extension of the internal language of a topos $\mathcal{T}_{\mathbf{Topos}}$ defined in [Mai05] with the inductive list constructor List of **MLTT** (in light of the results of [Mai05], the calculus $\mathcal{T}_{\mathbf{ArithTopos}}$ provides the internal language of toposes equipped with a natural number object).

Theorem 1.8.3. $\mathcal{T}_{\mathbf{ArithTopos}}$ *is a compatible extension of* **emTT** *up to equivalence of theories.*

Proof. Consider the extension of **emTT** with rules for enforcing impredicativity, and with a rule making propositions coincide with mono-types, that is types with at most one element.

$$\frac{A\ col}{A\ set} \qquad \frac{A\ prop}{A\ prop_s} \qquad \frac{x = y \in A\ col\ [x \in A, y \in A]}{A\ prop}$$

To show that the theory obtained in this way is equivalent to $\mathcal{T}_{\mathbf{ArithTopos}}$, we first interpret $\mathcal{P}(1)$ as the type classifier Ω; then, we need to check that $\mathcal{T}_{\mathbf{ArithTopos}}$ encodes the propositional constructors (except that of extensional propositional equality, which is already present) and the set constructors of the empty set, the disjoint sum, and the quotient set. For propositional constructors we can use the standard impredicative encodings; for the mentioned set constructors we can use the following encodings.

$$\begin{aligned} \mathsf{N_0} &:\equiv (\Pi\, U \in \mathcal{P}(1))\mathsf{T}(U) \\ A + B &:\equiv (\Sigma\, U \in \mathcal{P}(A))(\Sigma\, V \in \mathcal{P}(B))((\exists x \in A)(U =_{\mathcal{P}(A)} \{x\} \wedge V =_{\mathcal{P}(B)} \emptyset) \\ &\qquad\qquad \vee (\exists y \in B)(U =_{\mathcal{P}(A)} \emptyset \wedge V =_{\mathcal{P}(B)} \{y\})) \\ A/R &:\equiv (\Sigma\, U \in \mathcal{P}(A))(\exists x \in A)\, U =_{\mathcal{P}(A)} \{y \in A \,|\, R(x, y)\} \end{aligned}$$

To check that the above three definitions correctly interprets the corresponding constructor, it is vital to use the axiom of unique choice granted by the identification of propositions as mono-types.

Compatibility is straightforward since the way in which a topos with a natural number objects interprets the logic of $\mathbf{HA}^\omega$ is exactly through monomorphisms, which are reflected in its internal language by mono-types.

□

Theorem 1.8.4. **CZF**, *and thus* **IZF** *and* **ZF**, *are compatible extensions of* **emTT** *up to equivalence of theories.*

Proof. Proven in [MS22]. □

Theorem 1.8.5. *Both* **mTT** *and* **emTT** *are compatible with* **HoTT**.

Proof. Proven in [CM22]. □

Chapter 2

Inductive and Coinductive Predicates

Chapter Abstract

In this chapter, we show that in various foundations, predicates defined using wellfounded trees and those defined using non-wellfounded trees have topological counterparts in terms of inductively generated formal covers and coinductively generated positivity relations, respectively. As a corollary, we extend the compatibility results of both levels of the Minimalist Foundation to such constructions.

Although presented in parallel, the inductive part is adapted from joint work with M. E. Maietti [MS23a], while the coinductive part is from the work in [Sab24]. In this chapter, all the proofs performed within Martin-Löf's type theory have been checked in Agda. The source code is available on
https://github.com/PietroSabelli/topological-co-induction.

2.1 Overview

Starting with natural numbers, inductive definitions are pervasive in mathematics. They are even more so in predicative mathematics since, for example, the common way of defining a subset as the smallest one satisfying certain clauses must be replaced with an inductive definition. Perhaps less common are their dual: coinductive definitions. On the one hand, as it often happens in classical mathematics, this dualism is collapsed since coinductive definitions can be reduced to inductive ones with the Law of Excluded Middle (see Remark 2.2.5); on the other hand, contrary to induction, in constructive mathematics based on type theory the computational status of coinductive types has still not been settled.

As already mentioned, Formal Topology serves as a benchmark for predicative constructive mathematics (see Section 1.1.1). Indeed, in Formal Topology the need for both inductive and coinductive definitions strongly manifest itself; in particular, to represent the collections of open and closed subsets of many natural topologies. Powerful techniques for inductively generating the collection of open subsets and coinductively generating the one of closed subsets have been developed in [CSSV03; Sam19] and have since been a cornerstone of the field. The (co)inductive methods of Formal Topology were later implemented in the Minimalist Foundation in the form of an inductive propositional constructor and a coinductive one in [MMR22], where the authors also extended the quotient interpretation and the realisability interpretation of **MF** to account for them.

In this chapter, we discuss another way of extending the Minimalist Foundation with general-purpose inductive and coinductive definitions in the style of Aczel [Acz77]. The aim is twofold: on the one hand, we establish an equivalence between such (co)inductive methods and the ones of Formal Topology; on the other, we compare them to other established schemes of (co)induction in the literature, in particular with W-types and M-types of Martin-Löf's type theory and its extensions, most notably Homotopy Type Theory [ACS15]. Our final goal (Corollary 2.6.1) is to extend the compatibility results of the Minimalist Foundation once (co)inductive methods are added to it.

2.2 (Co)Inductive predicates in emTT

The basic idea behind any (co)inductive construction is to generate an object by proceeding from a given set of rules – intuitively, with induction following them in the forward direction and coinduction backwards. How such rules are specified and used heavily depends on *(1)* the kind of mathematical object to be constructed, and *(2)* the setting in which this construction is formalised.

Concerning the first point, a key distinction is between the (co)inductive generation of sets and the (co)inductive generation of predicates (or, equivalently, subsets); this distinction has been first put forward in the case of induction by Kleene in [Kle52], where he called the former *fundamental inductive definitions*, and the latter *non-fundamental inductive definitions*; the present section focuses on the latter. Paradigmatic examples of inductively and coinductively defined sets are lists (of which natural numbers are the most basic instance) and streams, respectively. On the other hand, the fundamental example of (co)inductive predicates we would consider is given by deductive systems, which, through their rules, for a given set of syntactical expressions declare inductively which are the derivable judgements and coinductively which are the refutable judgements.

Concerning the second point, each foundational theory has its peculiar ways of implementing (co)induction, e.g. (non-)wellfounded trees in Martin-Löf's type theories, Higher Inductive Types in Homotopy Type Theory, or Generalized Inductive Definitions in set theory. We are interested in formalising in dependent type theories the latter scheme, first introduced in [Acz77] and then adapted to a constructive setting in [Rat05; AR10]. To this end, we first define it in both levels of the Minimalist Foundation, starting in this section with the extensional one; then, we will show how to define its proof-relevant version in Martin-Löf's type theory.

The starting point is to specify how to declare rules for the (co)inductive generation of predicates. We do so using a notion that goes under two names, corresponding to two different – although related – interpretations. The first name is *rule set*[1], and, in this sense, it is a straightforward adaptation in the setting of the Minimalist Foundation of the homonym notion in [Acz77]. The second name is *axiom set*, defined in the context of (co)inductive generation of formal topologies in [CSSV03]. We will focus on the first interpretation (and hence use the first name) while presenting (co)inductive predicates. We will turn to the second name when recalling (co)inductive methods in formal topology.

Definition 2.2.1. A *rule* or *axiom set* over a given set A consists of the following data:

1. a dependent family of sets $I(x)$ *set* $[x \in A]$;
2. a dependent family of A-subsets $C(x, y) \in \mathcal{P}(A)$ $[x \in A, y \in I(x)]$.

Given two elements $a \in A$ and $i \in I(a)$, we say that i is a *rule* with *premises* $C(a, i)$ and *conclusion* a; sometimes it is represented pictorially as

$$\frac{C(a,i)}{a}\; i$$

Hopefully, the chosen terminology makes the logical interpretation of a rule set as an internally defined deduction system transparent. For the rest of this section, suppose to have fixed a set A and a rule set (I, C) over it.

Recall that a predicate P on A is just a proposition depending on A

$$P(x)\ \mathit{prop}\ [x \in A]$$

A rule set induces two ways of transforming predicates.

Definition 2.2.2. Given a predicate P on A, we define two other predicates $\mathsf{Der}_{I,C}(P)$ and $\mathsf{Conf}_{I,C}(P)$ on A as follows:

$$\mathsf{Der}_{I,C}(P)(x) :\equiv (\exists y \in I(x))(\forall z \,\varepsilon\, C(x,y))P(z)\ \mathit{prop}\ [x \in A]$$
$$\mathsf{Conf}_{I,C}(P)(x) :\equiv (\forall y \in I(x))(\exists z \,\varepsilon\, C(x,y))P(z)\ \mathit{prop}\ [x \in A]$$

[1] In [Rat05], the name is changed to *inductive definition*; we stick to the original terminology since it helps intuition, and our treatment is not limited to induction.

We call them *derivability* and *confutability from* P, respectively.

As the names suggest, derivability from P tells which elements of A can be derived with exactly one rule application, assuming as axioms the elements for which P holds – in fact, the definition of $\mathsf{Der}_{I,C}(P)(x)$ explicitly reads as *there exists a rule with conclusion x such that all its premises are satisfied by* P. Dually, confutability from P tells which elements of A can be confuted after exactly one step of backward search, assuming that the elements for which P holds are already refuted – the definition of $\mathsf{Conf}_{I,C}(P)(x)$ reads *all rules with conclusion x have at least one premise for which P holds.*

In accordance with the above interpretation, the constructions $\mathsf{Der}_{I,C}(-)$ and $\mathsf{Conf}_{I,C}(-)$ can be read meta-theoretically as two endomorphisms of the preorder of predicates on A: the preorder relation is formally given by

$$P \leq_A Q :\equiv (\forall x \in A)(P(x) \Rightarrow Q(x))\ prop$$

and it is straightforward to check that $P \leq_A Q$ implies both

$$\mathsf{Der}_{I,C}(P) \leq_A \mathsf{Der}_{I,C}(Q) \quad \text{and} \quad \mathsf{Conf}_{I,C}(P) \leq_A \mathsf{Conf}_{I,C}(Q).$$

Finally, notice that if P is a small proposition, then also $\mathsf{Der}_{I,C}(P)$ and $\mathsf{Conf}_{I,C}(P)$ are. Thanks to these observations, we can exploit the theory and terminology of monotone operators.

Definition 2.2.3. Following the terminology in [Rat05], we say that a predicate P on A is (I, C)-*closed* if it is closed with respect to $\mathsf{Der}_{I,C}(-)$, namely if

$$\mathsf{Der}_{I,C}(P) \leq_A P\ true$$

Dually, we say that P is (I, C)-*correct* if it is correct with respect to $\mathsf{Conf}_{I,C}(-)$, namely if

$$P \leq_A \mathsf{Conf}_{I,C}(P)\ true$$

Observe that, thanks to monotonicity, if P is a (I, C)-closed predicate, we have the following chain of inequalities.

$$\cdots \leq_A \mathsf{Der}_{I,C}(\mathsf{Der}_{I,C}(P)) \leq_A \mathsf{Der}_{I,C}(P) \leq_A P$$

Dually, if P is (I, C)-correct, it follows

$$P \leq_A \mathsf{Conf}_{I,C}(P) \leq_A \mathsf{Conf}_{I,C}(\mathsf{Conf}_{I,C}(P)) \leq_A \cdots$$

Therefore, being a closed predicate means that no new conclusion can be derived from it. Dually, to be a correct predicate means that if the elements satisfying it are assumed not to be derivable by the deduction system, then they will always be considered so even after any number of backward search steps. We are then naturally led to interpret the smallest closed predicate as expressing derivability and the greatest correct predicate as confutability. In the base theory, their existence is not guaranteed, let alone their smallness as propositions. Therefore, we will postulate them. Formally, this is done by introducing two new logical constructors Ind and CoInd. We now report their precise rules, together with the judgements that formalise the rule set; the latter will then be left implicit in the premises of the former.

Rule set parameters in emTT

$$A\ set \quad I(x)\ set\ [x \in A] \quad C(x,y) \in \mathcal{P}(A)\ [x \in A, y \in I(x)]$$

Rules for inductive predicates in emTT

$$\textsf{F-Ind}\ \frac{a \in A}{\mathsf{Ind}_{I,C}(a)\ prop_s}$$

$$\textsf{I-Ind}\ \frac{}{\mathsf{Der}_{I,C}(\mathsf{Ind}_{I,C}) \leq_A \mathsf{Ind}_{I,C}\ true}$$

$$\textsf{E-Ind}\ \frac{P(x)\ prop\ [x \in A] \quad \mathsf{Der}_{I,C}(P) \leq_A P\ true}{\mathsf{Ind}_{I,C} \leq_A P\ true}$$

Rules for coinductive predicates in emTT

$$\textsf{F-CoInd}\ \frac{a \in A}{\mathsf{CoInd}_{I,C}(a)\ prop_s}$$

$$\textsf{E-CoInd}\ \frac{}{\mathsf{CoInd}_{I,C} \leq_A \mathsf{Conf}_{I,C}(\mathsf{CoInd}_{I,C})\ true}$$

$$\textsf{I-CoInd}\ \frac{P(x)\ prop\ [x \in A] \quad P \leq_A \mathsf{Conf}_{I,C}(P)\ true}{P \leq_A \mathsf{CoInd}_{I,C}\ true}$$

Notice how, again, consistently with the intended interpretation, we can deduce from their rules that $\mathsf{Ind}_{I,C}$ and $\mathsf{CoInd}_{I,C}$ are fixed points for $\mathsf{Der}_{I,C}$ and $\mathsf{Conf}_{I,C}$, respectively.

Example 2.2.4. Recall that in Section 1.7.9 we defined the transitive closure R^+ of a relation R on a set A using the list constructor. Alternatively, we can define it as the inductive predicate over $A \times A$ defined through the rules

$$\frac{R(a,b)\ true}{R^+(a,b)} \qquad \frac{R^+(a,b) \quad R^+(b,c)}{R^+(a,b)}$$

which can be formalised using the following rule set.

$$\begin{aligned} I(\langle a,b\rangle) &:\equiv R(a,b) + A \\ C(\langle a,b\rangle, \mathsf{inl}(p)) &:\equiv \emptyset \\ C(\langle a,c\rangle, \mathsf{inr}(b)) &:\equiv \{\langle a,b\rangle, \langle b,c\rangle\} \end{aligned}$$

Notice that for this definition $R^+(x,y) :\equiv \mathsf{Ind}_{I,C}(\langle x,y\rangle)\ [x \in A, y \in A]$ we do not need to use the list constructor.

On the other hand, consider the following relation generated using the coinductive predicate on the same rule set.

$$\mathsf{Disc}_R(x,y) :\equiv \mathsf{CoInd}_{I,C}(\langle x,y\rangle)\ [x \in A, y \in A]$$

Interpreting R as a directed graph with A as its set of vertices, the predicate Disc_R positively expresses the fact that two elements of A are disconnected in it. Notice that proving $\mathsf{Disc}_R(a,b)$ for two specific terms $a, b \in A$ amounts to prove

$$\neg R(a,b) \wedge (\forall x \in A)(\mathsf{Disc}_R(a,x) \vee \mathsf{Disc}_R(x,b))$$

which is computationally more informative then the mere negation $\neg R^+(a,b)$.

Remark 2.2.5. Classically, i.e. assuming the Law of Excluded Middle, inductive and coinductive predicates are complementary subsets and thus mutually encodable as

$$\begin{aligned} \mathsf{CoInd}_{I,C}(x) &:\equiv \neg\mathsf{Ind}_{I,C}(x)\ prop\ [x \in A] \\ \mathsf{Ind}_{I,C}(x) &:\equiv \neg\mathsf{CoInd}_{I,C}(x)\ prop\ [x \in A] \end{aligned}$$

To see it, observe that the following equivalences between predicates hold.

$$\begin{aligned} \neg\mathsf{Der}_{I,C}(P)(x) &\Leftrightarrow \mathsf{Conf}_{I,C}(\neg P)(x)\ true\ [x \in A] \\ \neg\mathsf{Conf}_{I,C}(P)(x) &\Leftrightarrow \mathsf{Der}_{I,C}(\neg P)(x)\ \ true\ [x \in A] \end{aligned}$$

Therefore, a predicate P is (I, C)-closed (resp. (I, C)-correct) if and only if its complement $\neg P$ is (I, C)-correct (resp. (I, C)-closed) – and the smallest (I, C)-closed predicate is the complement of the greatest (I, C)-correct one, and vice versa.

On another note, both inductive and coinductive predicates are impredicatively encodable in the calculus $\mathcal{T}_{\mathbf{Topos}}$. This is done in the usual way, interpreting them as the intersection of all (I, C)-closed predicates and the union of all (I, C)-correct predicates, respectively.

$$\mathsf{Ind}_{I,C}(x) :\equiv (\forall P \in \mathcal{P}(A))(\mathsf{Der}_{I,C}(P) \leq_A P \Rightarrow x \,\varepsilon\, P) \; prop \; [x \in A]$$
$$\mathsf{CoInd}_{I,C}(x) :\equiv (\exists P \in \mathcal{P}(A))(P \leq_A \mathsf{Conf}_{I,C}(P) \wedge x \,\varepsilon\, P) \; prop \; [x \in A]$$

We end the presentation of (co)inductive predicates in the extensional level by presenting alternative introduction and elimination rules for them, in which some internal quantifiers are externalised. Although its mathematical content is perhaps less compact, they have the advantage, on the one hand, of emphasising the characters of introduction and elimination, thus being more suitable to design their intensional counterparts, and, on the other hand, of being more similar to the rules for the (co)inductive generation of formal topologies, and thus being more easily comparable with them.

Lemma 2.2.6. *The introduction and elimination rules for inductive predicates in* **emTT** *can equivalently be formulated in the following way.*

$$\mathsf{I\text{-}Ind'}\;\frac{a \in A \quad i \in I(a) \quad (\forall x \,\varepsilon\, C(a,i))\mathsf{Ind}_{I,C}(x) \; true}{\mathsf{Ind}_{I,C}(a) \; true}$$

$$\mathsf{E\text{-}Ind'}\;\frac{\begin{array}{l} P(x) \; prop \; [x \in A] \\ P(x) \; true \; [x \in A, y \in I(x), w \in (\forall z \,\varepsilon\, C(x,y))P(z)] \\ a \in A \quad \mathsf{Ind}_{I,C}(a) \; true \end{array}}{P(a) \;\; true}$$

Analogously, for coinductive predicates.

$$\mathsf{E\text{-}CoInd'}\;\frac{a \in A \quad i \in I(a) \quad \mathsf{CoInd}_{I,C}(a) \; true}{(\exists x \,\varepsilon\, C(a,i))\mathsf{CoInd}_{I,C}(x) \; true}$$

$$\mathsf{I\text{-}CoInd'}\;\frac{\begin{array}{l} P(x) \; prop \; [x \in A] \\ (\exists z \,\varepsilon\, C(x,y))P(z) \; true \; [x \in A, y \in I(x), w \in P(x)] \\ a \in A \quad P(a) \; true \end{array}}{\mathsf{CoInd}_{I,C}(a) \; true}$$

Proof. We show how to prove that $\mathsf{I\text{-}Ind}$ and $\mathsf{I\text{-}Ind}'$ are equivalent, the other equivalences being proven analogously.

First, notice that the rule $\mathsf{I\text{-}Ind}$ explicitly reads

$$(\forall x \in A)((\exists y \in I(x))(\forall z \,\varepsilon\, C(x,y))\mathsf{Ind}_{I,C}(z) \Rightarrow \mathsf{Ind}_{I,C}(x))\ true \qquad (2.1)$$

First, let us assume that 2.1 and the premises of $\mathsf{I\text{-}Ind}'$ hold. To derive the conclusion of $\mathsf{I\text{-}Ind}'$, apply the rule $\mathsf{E\text{-}}\forall$ to the universal statement of 2.1 and on the premise $a \in A$; then apply $\mathsf{E\text{-}} \Rightarrow$ to the resulting conditional statement and to a proof of the antecedent obtained with the rule $\mathsf{I\text{-}}\exists$ applied to the premisses $i \in I(a)$ and $\mathsf{true} \in (\forall x \,\varepsilon\, C(a,i))\mathsf{Ind}_{I,C}(x)$.

On the other hand, suppose that the rule $\mathsf{I\text{-}Ind}'$ holds; it obviously derives the judgment

$$\mathsf{Ind}_{I,C}(x)\ true\ [x \in A, y \in I(x), p \in (\forall z \,\varepsilon\, C(x,y))\mathsf{Ind}_{I,C}(z)]$$

One then proceeds by applying successively the rules $\mathsf{E\text{-}}\exists$, $\mathsf{I\text{-}} \Rightarrow$, and $\mathsf{I\text{-}}\forall$ to obtain 2.1. □

2.3 (Co)Inductive predicates in mTT

We now introduce the intensional counterparts of the rules for inductive and coinductive predicates. They are crucial, on the one hand, to extend the Minimalist Foundation's intensional level and have it interpret extensional (co)induction in its quotient model; on the other hand, to be able to compare our notion of (co)induction with that of other intensional theories.

A rule set is formalised similarly, except for the usual replacement of subsets for functions towards the universe of small propositions.

Rule set parameters in mTT

$$A\ set \quad I(x)\ set\ [x \in A] \quad C(x,y) \in A \to \mathsf{Prop_s}\ [x \in A, y \in I(x)]$$

The rules are then easily obtained from the extensional ones presented as in Lemma 2.2.6 by adding the now-relevant proof terms. Regarding the computational behaviour of the constructors, on the one hand, we can give computational meaning to inductive proofs drawing on the usual pattern of inductive types in Martin-Löf's type theory; on the other hand, we stayed close to the choice in [MMR22] of not postulating any computation rule for coinduction; we intend, as future works, to consider the addition of such rules, following their description in [Gim96].

Again, we keep the rule set parameters implicit in the premises.

Rules for inductive predicates in mTT

$$\textsf{F-Ind}\ \frac{a \in A}{\textsf{Ind}_{I,C}(a)\ prop_s}$$

$$\textsf{I-Ind}\ \frac{a \in A \quad i \in I(a) \quad p \in (\forall x\, \varepsilon\, C(a,i))\textsf{Ind}_{I,C}(x)}{\textsf{ind}(a,i,p) \in \textsf{Ind}_{I,C}(a)}$$

$$\textsf{E-Ind}\ \frac{\begin{array}{l} P(x)\ prop\ [x \in A] \\ c(x,y,w) \in P(x)\ [x \in A, y \in I(x), w \in (\forall z\, \varepsilon\, C(x,y))P(z)] \\ a \in A \quad p \in \textsf{Ind}_{I,C}(a) \end{array}}{\textsf{El}_{\textsf{Ind}}(a,p,(x,y,w).c) \in P(a)}$$

$$\textsf{C-Ind}\ \frac{\begin{array}{l} P(x)\ prop\ [x \in A] \\ c(x,y,w) \in P(x)\ [x \in A, y \in I(x), w \in (\forall z\, \varepsilon\, C(a,i))P(z)] \\ a \in A \quad i \in I(a) \quad p \in (\forall x\, \varepsilon\, C(a,i))\textsf{Ind}_{I,C}(x) \end{array}}{\textsf{El}_{\textsf{Ind}}(a,\textsf{ind}(a,i,p),(x,y,w).c) = c(a,i,\lambda z.\lambda q.\textsf{El}_{\textsf{Ind}}(z,p(z,q),(x,y,w).c)) \in P(a)}$$

Rules for coinductive predicates in mTT

$$\textsf{F-CoInd}\ \frac{a \in A}{\textsf{CoInd}_{I,C}(a)\ prop_s}$$

$$\textsf{E-CoInd}\ \frac{a \in A \quad i \in I(a) \quad p \in \textsf{CoInd}_{I,C}(a)}{\textsf{El}_{\textsf{CoInd}}(a,i,p) \in (\exists x\, \varepsilon\, C(a,i))\textsf{CoInd}_{I,C}(x)}$$

$$\textsf{I-CoInd}\ \frac{\begin{array}{l} P(x)\ prop\ [x \in A] \\ c(x,y,w) \in (\exists z\, \varepsilon\, C(x,y))P(z)\ [x \in A, y \in I(x), w \in P(x)] \\ a \in A \quad p \in P(a) \end{array}}{\textsf{coind}(a,p,(x,y,w).c) \in \textsf{CoInd}_{I,C}(a)}$$

Proposition 2.3.1. *Inductive and coinductive predicates are encodable in the Calculus of Constructions.*

Proof. The Calculus of Constructions can be regarded as an impredicative version of **mTT**. Thus, the proof adapts the same idea of Remark 2.2.5. We only need to make explicit the proof terms and, regarding the inductive predicate constructor, additionally show that the conversion rule is satisfied.

$$
\begin{aligned}
\mathsf{Ind}_{I,C}(a) &:\equiv (\forall P \in A \to \mathsf{Prop}) \\
&\qquad ((\forall x \in A)(\forall y \in I(x))((\forall z \,\varepsilon\, C(x,y))P(z) \Rightarrow P(x)) \\
&\qquad \Rightarrow P(a)) \\
\mathsf{ind}(a,i,p) &:\equiv \lambda P.\lambda c.c(a,i,\lambda x.\lambda q.p(x,q,P,c)) \\
\mathsf{El}^{P}_{\mathsf{Ind}}(a,p,(x,y,w).c) &:\equiv p(\lambda x.P(x), \lambda x.\lambda y.\lambda w.c(x,y,w))
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{CoInd}_{I,C}(a) &:\equiv (\exists P \in A \to \mathsf{Prop}) \\
&\qquad ((\forall x \in A)(P(x) \Rightarrow (\forall y \in I(x))(\exists z \,\varepsilon\, C(x,y))P(z)) \\
&\qquad \wedge P(a)) \\
\mathsf{El}_{\mathsf{CoInd}}(a,i,p) &:\equiv \mathsf{El}_{\exists}(p,(P,q).\mathsf{El}_{\exists}(c(a,r,i),(z,s).\langle z,\langle \pi_1(s),\langle P,c,\pi_2(s)\rangle\rangle\rangle)) \\
&\quad \text{where } c :\equiv \pi_1(q) \quad r :\equiv \pi_2(q) \\
\mathsf{coind}^{P}(a,p,(x,y,w).c) &:\equiv \langle \lambda x.P(x), \lambda x.\lambda y.\lambda w.c(x,y,w), p\rangle
\end{aligned}
$$

□

The extension of the quotient model and the realisability interpretation for the two new constructors will follow from the results already obtained for topological (co)induction once we establish their equivalence in the next section.

2.4 Topological (co)induction in MF

The development of the Minimalist Foundation has always been closely tied to that of Formal Topology, and it is no surprise that (co)induction has been first considered in that context. In two successive papers [MMR21; MMR22], the authors proposed extensions of the Minimalist Foundation supporting the formalisation of (co)inductive generation methods developed in Formal Topology. Moreover, they showed that both the setoid interpretation of the extensional level into the intensional one and the realisability interpretation of the latter can be adapted to support those extensions.

We now recall those extensions, starting with the basic definitions of Formal Topology formalised in the Minimalist Foundation.

Definition 2.4.1. A *basic topology* consists of the following data:

1. a set A, whose elements are called *basic opens*;
2. a small binary relation $\lhd$, called *basic cover*, between elements of A and subsets of A, satisfying the following properties:
   - *(reflexivity)* if $a \,\varepsilon\, V$, then $a \lhd V$;
   - *(transitivity)* if $a \lhd U$ and $(\forall x \,\varepsilon\, U) x \lhd V$, then $a \lhd V$.
3. a small binary relation $\ltimes$, called *positivity relation*, between elements of A and subsets of A, satisfying the following properties:
   - *(coreflexivity)* if $a \ltimes V$, then $a \,\varepsilon\, V$;
   - *(cotransitivity)* if $a \ltimes U$ and $(\forall x \in A)(x \ltimes V \Rightarrow x \,\varepsilon\, U)$, then $a \ltimes V$;
   - *(compatibility)* if $a \ltimes V$ and $a \lhd U$, then $(\exists x \,\varepsilon\, V)(x \ltimes U)$.

The intuitive meaning of the above definition is the following: the set A, as its name suggests, is a base for the topology; the relation $a \lhd V$ means that the basic open a is covered by the family of basic opens V; and finally, $a \ltimes V$ means that there is a point of a all whose basic neighbourhoods belongs to V.

In [CSSV03], the authors devised a way to inductively generate a basic cover on a given set A, starting from an axiom set (I, C) over it. In this sense, an axiom set is the set-indexed family of axioms

$$a \lhd C(a, i) \quad \text{for each } a \in A \text{ and } i \in I(a)$$

The inductively generated basic cover is then the smallest one which satisfies them. Later, in [Sam03], this idea was dualised to coinductively generate the greatest positivity relation satisfying

$$a \ltimes C(a, i) \quad \text{for each } a \in A \text{ and } i \in I(a)$$

which, moreover, turns out to be compatible with the inductively generated basic cover on the same axiom set, and thus gives rise to a basic topology.

Example 2.4.2. Consider the set $\mathsf{List}(\mathbb{N})$, and the following axiom set over it.

$$\begin{aligned} I(s) &:\equiv \mathsf{N}_1 + (\Sigma l \in \mathsf{List}(\mathbb{N}))(\exists t \in \mathsf{List}(\mathbb{N}))[l, t] =_{\mathsf{List}(\mathbb{N})} s \\ C(s, \mathsf{inl}(\star)) &:\equiv \{\mathsf{cons}(s, n) \mid n \in \mathbb{N}\} \\ C(s, \mathsf{inr}(z)) &:\equiv \{\pi_1(z)\} \end{aligned}$$

where $[-,-]$ is the concatenation operator. The basic topology obtained by (co)induction from the above axiom set is the *Baire space topology.* In particular, in this situation, the positivity relation $a \ltimes_{I,C} V$ precisely states that there exists a *spread* containing a and contained in V; see [CS19]. This is an example of (co)inductive definition that cannot be formalised in the bare calculus **emTT** (see Proposition 2.1 of [MMR21]).

When the positivity relation $a \ltimes V$ is specialised to the case where $V = A$, one talks of the *positivity predicate* $\mathsf{Pos}(a) :\equiv a \ltimes A$; sometimes, coinductively generated positivity predicates have been considered on their own, as in [MV04], where the authors used them to constructively and predicatively prove the coreflection of locales in open locales.

These methods were formalised in the Minimalist Foundation in the form of an inductive constructor $\lhd$ in [MMR21] and a coinductive constructor $\ltimes$ in [MMR22], having both as parameters an axiom set (I, C) over a set A (formalised as in the case of (co)inductive predicates), and a subset $V \in \mathcal{P}(A)$ in the extensional level, or a propositional function $V \in A \to \mathsf{Prop_s}$ in the intensional one. We first recall their rules in the extensional level. Again, we leave the parameters implicit in the premises.

Rules for inductive basic covers in emTT

$$\mathsf{F}\text{-}\lhd \; \frac{a \in A}{a \lhd_{I,C} V \; prop_s}$$

$$\mathsf{I_{rf}}\text{-}\lhd \; \frac{a \,\varepsilon\, V \; true}{a \lhd_{I,C} V \; true}$$

$$\mathsf{I_{tr}}\text{-}\lhd \; \frac{a \in A \quad i \in I(a) \quad (\forall x \,\varepsilon\, C(a,i)) x \lhd_{I,C} V \; true}{a \lhd_{I,C} V \; true}$$

$$\mathsf{E}\text{-}\lhd \; \frac{\begin{array}{l} P(x) \; prop \; [x \in A] \\ (\forall x \in A)(x \,\varepsilon\, V \vee (\exists y \in I(x))(\forall z \,\varepsilon\, C(x,y)) P(z) \Rightarrow P(x)) \; true \\ a \in A \quad a \lhd_{I,C} V \; true \end{array}}{P(a) \; true}$$

Rules for coinductive positivity relations in emTT

$$\mathsf{F}\text{-}\ltimes \; \frac{a \in A}{a \ltimes_{I,C} V \; prop_s}$$

$$\mathsf{E_{corf}}\text{-}\ltimes\ \frac{a \ltimes_{I,C} V\ true}{a\,\varepsilon\, V\ true}$$

$$\mathsf{E_{cotr}}\text{-}\ltimes\ \frac{a \in A \quad i \in I(a) \quad a \ltimes_{I,C} V\ true}{(\exists x\,\varepsilon\, C(a,i))x \ltimes_{I,C} V\ true}$$

$$\mathsf{I}\text{-}\ltimes\ \frac{\begin{array}{l} P(x)\ prop\ [x \in A] \\ (\forall x \in A)(P(x) \Rightarrow x\,\varepsilon\, V \wedge (\forall y \in I(x))(\exists z\,\varepsilon\, C(x,y))P(z))\ true \\ a \in A \quad P(a)\ true \end{array}}{a \ltimes_{I,C} V\ true}$$

The above rules are very close to those of (co)inductive predicates presented as in Lemma 2.2.6, the only difference being the presence of the parameter V, which implies, for each of the two constructors, an additional (co)reflection rule and the additional condition $x\,\varepsilon\, V$ on the predicates involved in the universal properties. To consider the clause induced by the parameter V, we modify the derivability and confutability endomorphisms in the following way.

$$\mathsf{Der}_{I,C,V}(P)(x) :\equiv x\,\varepsilon\, V \vee \mathsf{Der}_{I,C}(P)(x)\ prop\ [x \in A]$$
$$\mathsf{Conf}_{I,C,V}(P)(x) :\equiv x\,\varepsilon\, V \wedge \mathsf{Conf}_{I,C}(P)(x)\ prop\ [x \in A]$$

The next lemma shows that indeed the predicates $- \lhd_{I,C} V$ and $- \ltimes_{I,C} V$ can be seen as the smallest closed predicate of $\mathsf{Der}_{I,C,V}$ and the greatest correct predicate of $\mathsf{Conf}_{I,C,V}$, respectively.

Lemma 2.4.3. *The introduction and elimination rules for extensional inductive basic covers in* **emTT** *can equivalently be formulated in the following way.*

$$\mathsf{I}\text{-}\lhd'\ \frac{}{\mathsf{Der}_{I,C,V}(- \lhd_{I,C} V) \leq_A - \lhd_{I,C} V\ true}$$

$$\mathsf{E}\text{-}\lhd'\ \frac{P(x)\ prop\ [x \in A] \quad \mathsf{Der}_{I,C,V}(P) \leq_A P\ true}{- \lhd_{I,C} V \leq_A P\ true}$$

Analogously for extensional coinductive positivity relations.

$$\mathsf{E}\text{-}\ltimes'\ \frac{}{- \ltimes_{I,C} V \leq_A \mathsf{Conf}_{I,C,V}(- \ltimes_{I,C} V)\ true}$$

$$\mathsf{I}\text{-}\ltimes'\ \frac{P(x)\ prop\ [x \in A] \quad P \leq_A \mathsf{Conf}_{I,C,V}(P)\ true}{P \leq_A - \ltimes_{I,C} V\ true}$$

Proof. The proof is entirely analogous to that of Lemma 2.2.6. □

The intensional rules are designed similarly. Again, the inductive basic cover has a computation rule, while the coinductive positivity relation does not.

Rules for inductive basic covers in mTT

$$\textsf{F-}\lhd\ \frac{a \in A}{a \lhd_{I,C} V\ prop_s}$$

$$\textsf{I}_{\textsf{rf}}\textsf{-}\lhd\ \frac{a \in A \quad r \in a\,\varepsilon\, V}{\textsf{rf}(a,r) \in a \lhd_{I,C} V}$$

$$\textsf{I}_{\textsf{tr}}\textsf{-}\lhd\ \frac{a \in A \quad i \in I(a) \quad p \in (\forall x\,\varepsilon\, C(a,i))x \lhd_{I,C} V}{\textsf{tr}(a,i,p) \in a \lhd_{I,C} V}$$

$$\textsf{E-}\lhd\ \frac{\begin{array}{l} P(x)\ prop\ [x \in A] \\ q_1(x,y) \in P(x)\ [x \in A, y \in x\,\varepsilon\, V] \\ q_2(x,y,w) \in P(x)\ [x \in A, y \in I(x), w \in (\forall z\,\varepsilon\, C(x,y))P(z)] \\ a \in A \quad p \in a \lhd_{I,C} V \end{array}}{\textsf{El}_{\lhd}(a,p,(x,y).q_1,(x,y,w).q_2) \in P(a)}$$

$$\textsf{C}_{\textsf{rf}}\textsf{-}\lhd\ \frac{\begin{array}{l} P(x)\ prop\ [x \in A] \\ q_1(x,y) \in P(x)\ [x \in A, y \in x\,\varepsilon\, V] \\ q_2(x,y,w) \in P(x)\ [x \in A, y \in I(x), w \in (\forall z\,\varepsilon\, C(x,y))P(z)] \\ a \in A \quad r \in a\,\varepsilon\, V \end{array}}{\textsf{El}_{\lhd}(a,\textsf{rf}(a,r),(x,y).q_1,(x,y,w).q_2) = q_1(a,r) \in P(a)}$$

$$\textsf{C}_{\textsf{tr}}\textsf{-}\lhd\ \frac{\begin{array}{l} P(x)\ prop\ [x \in A] \\ q_1(x,y) \in P(x)\ [x \in A, y \in x\,\varepsilon\, V] \\ q_2(x,y,w) \in P(x)\ [x \in A, y \in I(x), w \in (\forall z\,\varepsilon\, C(x,y))P(z)] \\ a \in A \quad i \in I(a) \quad p \in (\forall x\,\varepsilon\, C(a,i))x \lhd_{I,C} V \end{array}}{\textsf{El}_{\lhd}(a,\textsf{tr}(a,i,p),q_1,q_2) = q_2(a,i,\lambda z.\lambda w.\textsf{El}_{\lhd}(z,p(z,w),q_1,q_2) \in P(a)}$$

Rules for coinductive positivity relation in mTT

$$\mathsf{F}\text{-}\ltimes\ \frac{a \in A}{a \ltimes_{I,C} V\ prop_s}$$

$$\mathsf{E_{corf}}\text{-}\ltimes\ \frac{a \in A \quad p \in a \ltimes_{I,C} V}{\mathsf{corf}(a,p) \in a\,\varepsilon\,V}$$

$$\mathsf{E_{cotr}}\text{-}\ltimes\ \frac{a \in A \quad i \in I(a) \quad p \in a \ltimes_{I,C} V}{\mathsf{cotr}(a,i,p) \in (\exists x\,\varepsilon\,C(a,i))x \ltimes_{I,C} V}$$

$$\mathsf{I}\text{-}\ltimes\ \frac{\begin{array}{l} P(x)\ prop\ [x \in A] \\ q_1(x,y) \in x\,\varepsilon\,V\ [x \in A, y \in P(x)] \\ q_2(x,y,w) \in (\exists z\,\varepsilon\,C(x,y))P(z)\ [x \in A, y \in I(x), w \in P(x)] \\ a \in A \quad p \in P(a) \end{array}}{\mathsf{coind}(a,p,q_1,q_2) \in a \ltimes_{I,C} V}$$

The following result shows that the two flavours of (co)induction considered so far are equivalent.

Theorem 2.4.4. *In both levels of the Minimalist Foundation, inductive basic covers and inductive predicates are mutually encodable, and so are coinductive positivity relations and coinductive predicates. In particular, coinductive predicates coincide with positivity predicates.*

Proof. We first prove it for the extensional level. To see that topological (co)induction encodes (co)inductive predicates, it is enough to choose V to be irrelevant; indeed, notice that, for each axiom set (I,C) over A, the operator $\mathsf{Der}_{I,C}$ (resp. $\mathsf{Conf}_{I,C}$) is equivalent to the operator $\mathsf{Der}_{I,C,\emptyset}$ (resp. $\mathsf{Conf}_{I,C,A}$). It is trivial to check that the following interpretations satisfy the rules of (co)inductive predicates.

$$\mathsf{Ind}_{I,C}(x) :\equiv x \lhd_{I,C} \emptyset$$
$$\mathsf{CoInd}_{I,C}(x) :\equiv \mathsf{Pos}(x) \equiv x \ltimes_{I,C} A$$

On the other hand, assume to have an axiom set (I,C) over A, and a subset $V \in \mathcal{P}(A)$; to prove that inductive predicates encode basic inductive covers we define an enlarged rule set by encoding the additional reflexivity clause given by V.

$$I_V(x) :\equiv x\,\varepsilon\,V + I(x)$$
$$C_V(x, \mathsf{inl}(p)) :\equiv \emptyset$$
$$C_V(x, \mathsf{inr}(y)) :\equiv C(x,y)$$

where, formally, we set $C_V(x,y,z) :\equiv \mathsf{T}(\mathsf{El}_+(y,(w).[\bot],(w).[C(x,w,z)]))$. We claim that the following interpretation satisfy the rules of inductive basic covers.

$$x \lhd_{I,C} V :\equiv \mathsf{Ind}_{I_V,C_V}(x)$$

To show that coinductive predicates encode coinductive positivity relations, we define a restriction of both the set A and the axiom set (I,C) by comprehension on those elements for which V already holds.

$$\begin{aligned} A^V &:\equiv (\Sigma w \in A) w \,\varepsilon\, V \\ I^V(x) &:\equiv I(\pi_1(x)) \\ C^V(x,y) &:\equiv \{z \in A^V \mid \pi_1(z) \,\varepsilon\, C(\pi_1(x),y)\} \end{aligned}$$

We claim that the following interpretation satisfies the rules of coinductive positivity relations as presented in Proposition 2.4.3.

$$x \ltimes_{I,C} V :\equiv (\exists p \in x \,\varepsilon\, V)\mathsf{CoInd}_{I^V,C^V}(\langle x,p\rangle)$$

We prove the soundness of the above encodings in the case of the intensional level, where, as an additional difficulty, we have to make explicit also the interpretation of the proof terms and, in the inductive case, to check that the computation rules are satisfied.

For the inductive case we set

$$\begin{aligned} \mathsf{rf}(a,r) &:\equiv \mathsf{ind}(a,\mathsf{inl}(r),\lambda x.\lambda y.\mathsf{El}_\bot(y)) \\ \mathsf{tr}(a,i,p) &:\equiv \mathsf{ind}(a,\mathsf{inr}(i),p) \\ \mathsf{El}_\lhd(a,p,q_1,q_2) &:\equiv \mathsf{El}_{\mathsf{Ind}}(a,p,(x,y,w).\mathsf{Ap}_\Rightarrow(f(x,y),w)) \end{aligned}$$

where $f(x,y) :\equiv \mathsf{El}_+(y,(u).\lambda w.q_1(x,u),(u).\lambda w.q_2(x,u,w))$. Notice in particular that the encoding of the elimination term $\mathsf{El}_\lhd$ has been performed by bearing in mind the corresponding computation rule it will need to satisfy, which is then trivially verified.

In the case of coinduction we have the following.

$$\begin{aligned} \mathsf{corf}(a,p) &:\equiv \mathsf{El}_\exists(p,(x,y).x) \\ \mathsf{cotr}(a,i,p) &:\equiv \mathsf{El}_\exists(d,(z,w).\langle \pi_1(z),\pi_1(w),\pi_2(z),\pi_2(w)\rangle) \end{aligned}$$

where the d is a shorthand for the proof-term $\mathsf{El}_{\mathsf{CoInd}}(\langle a,\mathsf{El}_\exists(p,(x,y).x)\rangle,i,\mathsf{El}_\exists(p,(x,y).y))$ of the proposition $(\exists z \in (\Sigma x \in A)x \,\varepsilon\, V)(C(a,i,\pi_1(z)) \wedge \mathsf{CoInd}_{I^V,C^V}(z))$; for the coinduction term, assume to have a predicate $P(x)$ *prop* $[x \in A]$ and terms q_1 and q_2 as in the premise of the rule $\mathsf{I}\text{-}\ltimes$; we define the auxiliary predicate

$$P'(x) :\equiv P(\pi_1(x)) \; prop \; [x \in A^V]$$

together with the auxiliary term

$$c'(x, y, w) \in (\exists z \in C^V(x, y))P'(z) \; [x \in A^V, y \in I^V(x), w \in P'(x)]$$
$$c'(z, y, w) :\equiv \mathsf{El}_\exists(q_2(\pi_1(z), y, w), (u, v).\langle\langle u, q_1(u, \pi_2(v))\rangle, \pi_1(v), \pi_2(v)\rangle)$$

finally, we can write the following proof-term

$$\mathsf{coind}(a, p, q_1, q_2) :\equiv \langle q_1(a, p), \mathsf{coind}(\langle a, q_1(a, p)\rangle, p, (x, y, w).c')\rangle$$

□

As a corollary, we obtain that both the quotient model and the realisability interpretation of the Minimalist Foundation extend to (co)inductive predicates.

In [MMR21], the (two-level) theory obtained by extending the Minimalist Foundation with inductive basic covers was called $\mathbf{MF}_{\mathsf{ind}}$; then, in [MMR22], the theory obtained by extending the Minimalist Foundation with both inductive basic covers and coinductive positivity relations was called $\mathbf{MF}_{\mathsf{cind}}$. Here, in light of Theorem 2.4.4, we overload the notation by using the same names $\mathbf{MF}_{\mathsf{ind}}$ and $\mathbf{MF}_{\mathsf{cind}}$ to refer to the theory obtained by extending the Minimalist Foundation with inductive predicates and with both inductive and coinductive predicates, respectively.

2.5 (Co)Induction in MLTT

Induction and coinduction in Martin-Löf's type theory and its extensions can assume many forms; the two paradigmatic schemes are given by a pair of dual constructions called W-types for induction and M-types for coinduction. In this and the next section, we examine their relationship with the forms of (co)induction introduced in the previous section.

In this chapter, we work with a version $\mathbf{ML}_1^\eta$ of intensional Martin-Löf's type theory [NPS90] with the following type constructors: the empty type N_0, the unit type N_1, dependent sums Σ, dependent products Π, the list constructor List, identity types Id, disjoint sums $+$, and a universe of small types U_0 à la Russell closed under all the above type constructors. Inductive type constructors are defined to allow elimination toward all (small and large) types; this will also be true for the inductive types introduced in the subsequent sections – this feature will be essential for defining predicates recursively. Moreover, we assume η-equalities for N_1, Σ-types, and Π-types. These weaker extensional assumptions are justified in an intensional context since they do not break any computational property, even being assumed by default in the Agda implementation of Martin-Löf's type theory.

Finally, we will also consider extending the theory with the axiom of function extensionality funext, which for twoin **MLTT** is rendered as

$$(\Pi f, g \in (\Pi x \in A)B(x))(\Pi x \in A)\mathsf{Id}(B(x), f(x), g(x)) \to \mathsf{Id}((\Pi x \in A)B(x), f, g)$$

and the additional elimination scheme for the identity type known as *axiom* K introduced in [Str93].

$$\textsf{E-K}\ \frac{\begin{array}{c}M(x,y)\ type\ [x \in A, y \in \mathsf{Id}(A,x,x)] \\ m(x) \in M(x, \mathsf{id}(x))\ [x \in A] \quad a \in A \quad p \in \mathsf{Id}(A,a,a)\end{array}}{\mathsf{K}(p, (x).m) \in M(a,p)}$$

$$\textsf{C-K}\ \frac{\begin{array}{c}M(x,y)\ type\ [x \in A, y \in \mathsf{Id}(A,x,x)] \\ m(x) \in M(x, \mathsf{id}(x))\ [x \in A] \quad a \in A\end{array}}{\mathsf{K}(\mathsf{id}(a), (x).m) = m(a) \in M(a, \mathsf{id}(a))}$$

Recall that from axiom K (together with the standard eliminator of Martin-Löf's identity type) one can derive a proof-term of the axiom of Uniqueness of Identity Proof.

$$\mathsf{Id}(\mathsf{Id}(A,x,y),p,q)\ type\ [x, y \in A, p, q \in \mathsf{Id}(A,x,y)] \qquad (\mathsf{UIP})$$

2.5.1 Induction in MLTT

In Martin-Löf's type theory, one of the main ways of generating inductive sets is through the W-*type* constructor, also known as the *wellfounded trees* constructor. The parameters of a W-type consist of a set A and an A-indexed family of sets B, which together are often referred to as a *container* [AGMM15]; the resulting type $\mathsf{W}_{A,B}$ is intuitively understood as the set of wellfounded trees with nodes labelled by elements of A and with a (possibly infinitary) branching function given by B. The precise rules of the constructor are reported below.

Rules for W-types in $\mathbf{ML}_1^\eta$

$$\text{F-W}\ \frac{}{\mathsf{W}_{A,B} \in \mathsf{U}_0}$$

$$\text{I-W}\ \frac{a \in A \qquad f \in B(a) \to \mathsf{W}_{A,B}}{\mathsf{sup}(a, f) \in \mathsf{W}_{A,B}}$$

$$\text{E-W}\ \frac{\begin{array}{l} M(w)\ \mathit{type}\ [w \in \mathsf{W}_{A,B}] \\ d(x, h, k) \in M(\mathsf{sup}(x, h))\ [x \in A\,, h \in B(x) \to \mathsf{W}_{A,B}\,, k \in (\Pi y \in B(x))M(h(y))] \\ t \in \mathsf{W}_{A,B} \end{array}}{\mathsf{El}_{\mathsf{W}}(t, (x, h, k).d) \in M(t)}$$

$$\text{C-W}\ \frac{\begin{array}{l} M(w)\ \mathit{type}\ [w \in \mathsf{W}_{A,B}] \\ d(x, h, k) \in M(\mathsf{sup}(a, f))\ [x \in A\,, h \in B(a) \to \mathsf{W}_{A,B}\,, k \in (\Pi y \in B(x))M(h(y))] \\ a \in A \quad f \in B(a) \to \mathsf{W}_{A,B} \end{array}}{\mathsf{El}_{\mathsf{W}}(\mathsf{sup}(a, f), (x, h, k).d) = d(a, f, \lambda y.\mathsf{El}_{\mathsf{W}}(f(y), (x, h, k).d)) \in M(\mathsf{sup}(a, f))}$$

However, W-types do not seem to have the same expressive power as inductive predicates. This is because they produce just plain sets, while predicates, under the propositions-as-types paradigm, are families of sets. This same limitation has been addressed in [PS89], where the authors proposed a generalization of W-types, called *dependent* W*-types*, capable of constructing *families of mutually* inductive sets; in the literature they are also known as *general trees*, or *indexed* W*-types*. Dependent W-types are interpreted again as sets of wellfounded trees with labelled nodes[2]; however, each label now has a set of possible options for the branching function; moreover, each branching function not only indicates the number of subtrees but also dictates how each of their roots is to be labelled. The formal rendering of this intuition in Martin-Löf's type theory goes as follows. The parameters of a dependent W-type, which, analogously to the non-dependent case, are referred to as *indexed container*, consist of a small type $A \in \mathsf{U}_0$ of nodes' labels and a family of sets $I \in A \to \mathsf{U}_0$ indexing the possible branching functions associated with each label. To formalise the branching functions, there are two possibilities. Either with a function

$$C(x, y) \in A \to \mathsf{U}_0\ [x \in A, y \in I(x)]$$

[2]For their interpretation as free term algebras for infinitary multi-sorted signatures, see [Emm21].

that for each label says how many immediate subtrees there are with roots labelled by it; or with two arity functions

$$Br(x, y) \in \mathsf{U_0}\ [x \in A, y \in I(x)]$$
$$ar(x, y) \in Br(x, y) \to A\ [x \in A, y \in I(x)]$$

that say how many immediate subtrees there are in general and the label of each subtree's root, respectively. It is clear how, in the first case, indexed containers correspond precisely to the notion of rule set formulated in Martin-Löf's type theory. In either case, the type constructors are then formalised by adapting the pattern of W-types to account for the extra indexing. In particular, using the first formulation, one obtains a proof-relevant version of inductive predicates in Martin-Löf's type theory; as always, their elimination rules differ from the ones in the Minimalist Foundation since now they can work towards sets depending on the constructor. For this reason, in the following discussion, we will refer to dependent W-types defined with the parameter C as *(proof-relevant) inductive predicates*, reserving the name *dependent* W*-types* to just those defined using the parameters Br and ar. Their precise rules are spelt out in the appendix A.3. Finally, once more following the pattern of dependent W-types, in [MMR21] the authors primitively introduced in Martin-Löf's type theory also a constructor $\lhd$ formalising proof-relevant inductive basic covers, whose rules are again recalled in the appendix A.3.

By the results of [MS23a], we know that all the inductive constructors presented so far can be reduced to W-types.

Theorem 2.5.2. *In* $\mathbf{ML}_1^\eta$*, the following type constructors are mutually encodable.*

1. W*-types;*
2. *dependent* W*-types;*
3. *inductive predicates;*
4. *inductive basic covers.*

Proof. In the following, we just show how to interpret types. The long-but-routine calculations lie in explicitly constructing the interpretations for terms and verifying, for the first statement, that the conversion rules are satisfied using η-equalities and for the second statement, that the two types are internally equivalent using function extensionality. We leave these details to the meticulous reader or the proof-checking software.

W*-types encode dependent* W*-types.* The question of reducing dependent W-types to W-types has been answered positively in a number of extensional settings [PS89; GH04; AGMM15]; in particular, the technique used in [AGMM15] has been verified in the intensional setting of the Coq proof-assistant [Hug17]; we recast it in our setting.

Assume to have the parameters of a dependent W-type (A, I, Br, ar). Firstly, we construct a W-type Free of wellfounded trees containing information on the labels and the branching options of the nodes of the dependent W-type trees we are trying to encode. We do so by saying that the nodes of Free are labelled by dependent pairs of a label $a \in A$ and a choice of branching $i \in I(a)$ for it, and that the branching function is chosen correspondently to the parameter Br. Formally, we are constructing the set

$$\mathsf{Free} :\equiv \mathsf{W}_{(\Sigma x \in A)I(x), Br(\pi_1(z), \pi_2(z))}$$

Secondly, we thin out the set of Free trees with a A-indexed family of predicates $\mathsf{Legal}(x) \in \mathsf{Free} \to \mathsf{U}_0 \; [x \in A]$ which assert, for a Free tree, that its root's label has x as its first component and that the root's label first component of each of its subtrees respects the arity function ar. The predicate is formally defined by recursion on W-types as

$$\mathsf{Legal}(a, \mathsf{sup}(\langle b, i \rangle, f)) :\equiv \mathsf{Id}(A, a, b) \times (\Pi z \in Br(b, i))\mathsf{Legal}(ar(b, i, z), f(z))$$

Note that, being a fixed, the identity type appearing inside the Legal statement has a unique proof propositionally, and hence it is a contractible type according to [Uni13].

Then, our candidate for encoding the dependent W-type is the following type family.

$$\mathsf{DW}_{Br,ar}(a) :\equiv (\Sigma w \in \mathsf{Free})\mathsf{Legal}(a, w)$$

We can straightforwardly define a introduction term for $\mathsf{DW}'(a)$ in the following way.

$$\mathsf{dsup}(a, i, f) :\equiv \langle\ \mathsf{sup}(\langle a, i \rangle, \lambda z.\pi_1(f(z)))\ ,\ \langle\ \mathsf{id}(a), \lambda z.\pi_2(f(z))\ \rangle\ \rangle$$

The elimination term is a bit more involved, although it is not conceptually harder. Suppose to have, as in the premises of the elimination rule, a type family $M(a, w)$ *type* $[a \in A, w \in \mathsf{DW}'(a)]$ and a dependent term

$$\begin{aligned} & d(a, i, f, h) \in M(a, \mathsf{dsup}'(a, i, f)) \\ & \quad [a \in A, \\ & \qquad i \in I(a), \\ & \qquad\quad f \in (\Pi z \in Br(a, i))\mathsf{DW}'(ar(a, i, z)), \\ & \qquad\qquad h \in (\Pi z \in Br(a, i))M(ar(a, i, z), f(z))] \end{aligned}$$

we want to define a dependent term

$$\mathsf{El}'_{\mathsf{DW}}(a, w, d) \in M(a, w) \; [a \in A, w \in \mathsf{DW}'(a)]$$

validating the computational rule, that is, satisfying the following definitional equality.

$$\begin{aligned} & \mathsf{El}'_{\mathsf{DW}}(a, \mathsf{dsup}'(a, i, f), d) = d(a, i, f, \lambda z.\mathsf{El}'_{\mathsf{DW}}(ar(a, i, f), f(z), d)) \\ & \quad [a \in A, \\ & \qquad i \in I(a), \\ & \qquad\quad f \in (\Pi z \in Br(a, i))\mathsf{DW}'(ar(a, i, z))] \end{aligned}$$

The idea is to define the term $\mathsf{El}'_{\mathsf{DW}}$ by recursion by mimicking the above requirement – so that the task of checking the computation rule will turn out to be trivial. The definition explicitly reads

$$\begin{aligned} & \mathsf{El}_{\mathsf{DW}}(a, \langle \mathsf{sup}(\langle a, i\rangle, f), \mathsf{id}(a), l\rangle, d) :\equiv \\ & \quad d(a, i, \lambda z.\langle f(z), l(z)\rangle, \lambda z.\mathsf{El}_{\mathsf{DW}}(ar(a, i, z), \langle f(z), l(z)\rangle, d)) \end{aligned}$$

where we have implicitly used the recursion principles of Σ-types, W-types and identity types, all at once. The long-but-routine calculations lie in checking that the given definitions are well-typed by formulating it only with eliminator terms. In particular, it is in this step of defining $\mathsf{El}'_{\mathsf{DW}}$ by recursion that η-equalities are needed to ensure that the calculations go through. We leave them to the assiduous reader or to the proof-checker.

(Dependent W*-types encode inductive predicates).* Given an axiom set (I, C) over A, we can obtain an indexed container by considering the following arity functions.

$$\begin{aligned} Br(a, i) &:\equiv (\Sigma x \in A)C(a, i, x) \\ ar(a, i) &:\equiv \lambda z.\pi_1(z) \end{aligned}$$

It is easy to check that the dependent W-type $\mathsf{DW}_{Br,ar}$ constructed with the above parameters encodes the inductive predicate $\mathsf{Ind}_{I,C}$.

(Inductive predicates encode inductive basic covers). For this point we start by reasoning similarly as in the Minimalist Foundation. We encode the additional reflexive clause through a new rule set.

$$\begin{aligned} I_V(a) &:\equiv V(a) + I(a) \\ C_V(a, \mathsf{inl}(p)) &:\equiv \lambda z.\mathsf{N}_0 \\ C_V(a, \mathsf{inr}(i)) &:\equiv C(a, i) \end{aligned}$$

As in the **MF** case, the induced inductive predicate $\mathsf{Ind}_{I_V,C_V}(a)$ can be shown to be logically equivalent to the inductive basic cover $a \lhd_{I,C} V$. However, the above interpretation is not enough, since it does not satisfy the elimination towards dependent sets. The problem is the same as that encountered when naively trying to encode natural numbers using W-types in an intensional setting (see [NPS90]), and was solved in [Hug21] by employing a technique we will apply to the present case. The trick is to impose on the reflexive cases an extra canonicity condition expressed by a family of predicates $\mathsf{Canonical}(x) \in \mathsf{Ind}_{I_V,C_V}(x) \to \mathsf{U}_0\ [x \in A]$ defined by recursion as follows.

$$\begin{aligned} \mathsf{Canonical}(a, \mathsf{ind}(a, \mathsf{inl}(p), f)) &:\equiv f =_{(\Pi x \in A)(\mathsf{N}_0 \to \mathsf{Ind}_{I_V,C_V}(x))} \lambda x.\lambda z.\mathsf{El}_{\mathsf{N}_0}(z) \\ \mathsf{Canonical}(a, \mathsf{ind}(a, \mathsf{inr}(i), f)) &:\equiv (\Pi x \in A)(\Pi y \in C(a, i, x))\mathsf{Canonical}(f(x, y)) \end{aligned}$$

The following can then be checked to be a working interpretation.

$$a \lhd_{I,C} V :\equiv (\Sigma w \in \mathsf{Ind}_{I_V,C_V}(a))\mathsf{Canonical}(a, w)$$

(Inductive basic covers encode W*-types).* Given a container (A, B), consider the axiom set over the singleton set N_1 given by $I(x) :\equiv A$ and $C(x, y, z) :\equiv B(y)$. The W-type $\mathsf{W}_{A,B}$ is then encoded by $\star \lhd_{I,C} \emptyset$. □

In the presence of function extensionality, the above constructors enjoy neat categorical semantics, which we now review since it will be vital next for treating coinduction.

The type $\mathsf{W}_{A,B}$ can be shown to be the support of an initial algebra for the so-called *polynomial endofunctor* $\mathsf{P}_{A,B}(X) :\equiv (\Sigma x \in A)(B(x) \to X)$ on the category of types [MP00]. Analogously, inductive predicates, inductive basic covers and dependent W-types enjoy categorical semantics as initial algebras of the following *dependent polynomial endofunctors* on the category of A-dependent types, respectively.

$$\begin{aligned}
\mathsf{Der}_{I,C}(P)(x) &:\equiv (\Sigma y \in I(x))(\Pi z \in A)(C(x,y,z) \to P(z)) \\
\mathsf{Der}_{I,C,V}(P)(x) &:\equiv V(x) + (\Sigma y \in I(x))(\Pi z \in A)(C(x,y,z) \to P(z)) \\
\mathsf{Der}_{Br,ar}(P)(x) &:\equiv (\Sigma y \in I(x))(\Pi z \in Br(x,y))P(ar(x,y,z))
\end{aligned}$$

We chose to reuse the name Der because the first endofunctor clearly is the interpretation under the propositions-as-types paradigm of the derivability constructor used to define inductive predicates in the Minimalist Foundation.

Remark 2.5.3. We recall that polynomial endofunctors of the form $\mathsf{Der}_{Br,ar}$ can be described in an arbitrary locally cartesian closed category $\mathcal{C}$ [GH04] as follows. Indexed containers are specified by diagrams of the form

$$A \xleftarrow{ar} Br \xrightarrow{br} I \xrightarrow{i} A$$

and the resulting endofunctor is obtained as the composition of the following functors

$$\mathcal{C}/A \xrightarrow{ar^*} \mathcal{C}/Br \xrightarrow{\Pi_{br}} \mathcal{C}/I \xrightarrow{\Sigma_i} \mathcal{C}/A$$

where ar^* is the pullback functor, Π_{br} is the dependent product functor induced by the locally cartesian closure of $\mathcal{C}$, and Σ_i is the coproduct functor, that is postcomposition by i.

The following lemma formally proves that, in the presence of function extensionality, the two choices we discussed for formalising the intuition behind a dependent well-founded tree are equivalent.

Lemma 2.5.4. *Each endofunctor* $\mathsf{Der}_{I,C}$ *on the category of A-dependent types of* $\mathbf{ML}_1^\eta + \mathsf{funext}$ *induced by a rule-set* (A, I, C) *is naturally isomorphic to an endofunctor of the form* $\mathsf{Der}_{Br,ar}$ *induced by the parameters* (Br, ar) *for a dependent* W*-type and viceversa.*

Proof. To prove the first statement assume a rule set (I, C) over a set A and consider the following parameters.

$$\begin{aligned}
Br(x,y) &:\equiv (\Sigma z \in A)C(x,y,z) \\
ar(x,y,w) &:\equiv \pi_1(w)
\end{aligned}$$

For the converse, assuming the parameters A, I, Br and ar, consider the axiom set (I, C), where

$$C(x, y, z) :\equiv (\Sigma w \in Br(x, y))\mathsf{Id}(A, ar(x, y, w), z)$$

In both cases, it is easy to write a natural isomorphism between the two induced endofunctors $\mathsf{Der}_{I,C}$ and $\mathsf{Der}_{Br,ar}$. □

Finally, notice that if we were interested just in the proof-irrelevant, logical semantics of those inductive constructors, we could regard the endofunctors above simply as endomorphism on the preorder reflection of the category of A-dependent sets and the type constructors just as their smallest fixed points.

2.5.5 Coinduction in MLTT

As mentioned, in Martin-Löf type theory, coinduction is usually considered through a construction dual to W-types, called M-*types* (also known as *non-wellfounded trees*). The parameters of M-types are the same as those of W-types; as their alternative name suggests, a type $\mathsf{M}_{A,B}$ is intuitively understood as the set of non(-necessarily)-wellfounded trees with nodes labelled by elements of A and with branching function given by B. However, M-types are not usually presented through explicit inference rules; instead, they are characterised semantically by a universal property dual to that of W-types, namely by being terminal coalgebras for polynomial endofunctors $\mathsf{P}_{A,B}$. Analogously, there also exist *dependent* M-*types*, dualising dependent W-types and defined as the terminal coalgebras of dependent polynomial endofunctors $\mathsf{Der}_{Br,ar}$ or, equivalently by Lemma 2.5.4, $\mathsf{Der}_{I,C}$. In these forms, it has been shown that plain and dependent M-types are encodable using W-types in Martin-Löf's type theory extended with function extensionality and axiom K [AAG05], and in Homotopy Type Theory [ACS15].

Here, to define coinductive predicates and positivity relations in Martin-Löf's type theory, we follow instead the axiomatic approach taken in [MMR22], where rules for proof-relevant coinductive positivity relations have been explicitly introduced in Martin-Löf's type theory by just taking the corresponding rules for the intensional level of the Minimalist Foundation after identifying propositions with sets.

Rules for coinductive predicates in $\mathbf{ML}_1^\eta$

$$\text{F-CoInd}\ \frac{}{\mathsf{CoInd}_{I,C} \in A \to \mathsf{U}_0}$$

$$\text{E-CoInd}\ \frac{a \in A \quad i \in I(a) \quad p \in \mathsf{CoInd}_{I,C}(a)}{\mathsf{El}_{\mathsf{CoInd}}(a,i,p) \in (\Sigma x \in A)(C(a,i,x) \times \mathsf{CoInd}_{I,C}(x))}$$

$$\text{I-CoInd}\ \frac{\begin{array}{l} M(x)\ type\ [x \in A] \\ d(x,y,w) \in (\Sigma z \in A)(C(x,y,z) \times M(z))\ [x \in A, y \in I(x), w \in M(x)] \\ a \in A \quad m \in M(a) \end{array}}{\mathsf{coind}(a,m,(x,y,w).d) \in \mathsf{CoInd}_{I,C}(a)}$$

Rules for coinductive positivity relations in $\mathbf{ML}_1^\eta$

$$\text{F-}\ltimes\ \frac{}{- \ltimes_{I,C} V \in A \to \mathsf{U}_0}$$

$$\text{E-corf-}\ltimes\ \frac{a \in A \quad p \in a \ltimes_{I,C} V}{\mathsf{corf}(a,p) \in V(a)}$$

$$\text{E-cotr-}\ltimes\ \frac{a \in A \quad i \in I(a) \quad p \in a \ltimes_{I,C} V}{\mathsf{cotr}(a,i,p) \in (\Sigma x \in A)(C(a,i,x) \times x \ltimes_{I,C} V)}$$

$$\text{I-}\ltimes\ \frac{\begin{array}{l} M(x)\ type\ [x \in A] \\ q_1(x,y) \in V(x)\ [x \in A, y \in M(x)] \\ q_2(x,y,w) \in (\Sigma z \in A)(C(x,y,z) \times x \ltimes_{I,C} V)\ [x \in A, y \in I(x), w \in M(x)] \\ a \in A \quad p \in M(a) \end{array}}{\mathsf{coind}(a,p,q_1,q_2) \in a \ltimes_{I,C} V}$$

Notice that in this way, coinductive predicates and positivity relations are interpreted as the greatest fixed point of the following *dependent copolynomial endofunctors* defined in Martin-Löf's type theory and viewed as endomorphisms on the preorder reflection of the category of A-dependent types, respectively.

$$\mathsf{Conf}_{I,C}(P)(x) :\equiv (\Pi y \in I(x))(\Sigma z \in A)(C(x,y,z) \times P(z))$$
$$\mathsf{Conf}_{I,C,V}(P)(x) :\equiv V(x) \times (\Pi y \in I(x))(\Sigma z \in A)(C(x,y,z) \times P(z))$$

Clearly, we also define, analogously to the inductive case, an alternative copolynomial endofunctor $\mathsf{Conf}_{Br,ar}$ induced by an indexed container (A, I, Br, ar) as

$$\mathsf{Conf}_{Br,ar}(P)(x) :\equiv (\Pi y \in I(x))(\Sigma z \in Br(x, y))P(ar(x, y, z))$$

We can derive the dual result relating the two versions.

Lemma 2.5.6. *Each endofunctor* $\mathsf{Conf}_{I,C}$ *on the category of* A*-dependent types of* $\mathbf{ML}_1^\eta + \mathsf{funext}$ *induced by a rule-set* (A, I, C) *is naturally isomorphic to an endofunctor of the form* $\mathsf{Conf}_{Br,ar}$ *induced by the parameters* (Br, ar) *for a dependent* W*-type and viceversa.*

Proof. The proof is identical to that of Lemma 2.5.4. □

Finally, we can give a purely categorical description also of the endofunctors $\mathsf{Con}_{Br,ar}$ in an arbitrary locally cartesian closed category $\mathcal{C}$ as the composition of the following functors

$$\mathcal{C}/A \xrightarrow{ar^*} \mathcal{C}/Br \xrightarrow{\Sigma_{br}} \mathcal{C}/I \xrightarrow{\Pi_i} \mathcal{C}/A$$

The way they are defined, coinductive predicates are not related to inductive predicates in the same way that M-types are related to W-types: while coinductive predicates are defined as the greatest fixed points of the operator Conf, which is itself dual to the defining operator Der for inductive predicates, M-types and W-types (and their dependent versions) are terminal coalgebras and initial algebras, respectively, of the *same* endofunctors. For this reason, it seems we cannot get a fully symmetrical result to the one obtained in the inductive case. Nonetheless, in Martin-Löf's type theory, thanks to the axiom of choice, we can prove that the class of constructors $\mathsf{Der}_{I,C}$ can encode the class $\mathsf{Conf}_{I,C}$.

Proposition 2.5.7. *Each endofunctor* $\mathsf{Conf}_{I,C}$ *on the category of* A*-dependent types of* $\mathbf{ML}_1^\eta$ *induced by a rule-set* (A, I, C) *is naturally isomorphic to an endofunctor of the form* $\mathsf{Der}_{Br,ar}$ *induced by the parameters* (Br, ar) *for a dependent* W*-type.*

Proof. Assume to have a rule set (I, C) over A, and consider the following parameters.

$$\begin{aligned} I'(x) &:\equiv (\Pi y \in I(x))(\Sigma z \in A)C(x, y, z) \\ Br(x, f) &:\equiv I(x) \\ ar(x, f, y) &:\equiv \pi_1(f(y)) \end{aligned}$$

For each A-dependent type $P(x)$ *type* $[x \in A]$, we obtain the following isomorphism between A-dependent types.

$$\begin{aligned}
\mathsf{Der}_{Br,ar}(P)(x) &= (\Sigma f \in (\Pi y \in I(x))(\Sigma z \in A)C(x,y,z))(\Pi y \in I(x))P(\pi_1(f(y))) \\
&\cong (\Pi y \in I(x))(\Sigma w \in (\Sigma z \in A)C(x,y,z))P(\pi_1(w)) \\
&\cong (\Pi y \in I(x))(\Sigma z \in A)(C(x,y,z) \times P(z)) \\
&= \mathsf{Conf}_{I,C}(P)(x)
\end{aligned}$$

Notice that the first isomorphism above is precisely an application of the type-theoretic axiom of choice. It is easy to check that such family of isomorphisms are natural in P. □

Corollary 2.5.8. *Coinductive predicates and coinductive positivity relations are mutually encodable in* $\mathbf{ML}_1^\eta$*. Moreover, they are both encodable in any theory extending* $\mathbf{ML}_1^\eta$ *in which the greatest fixed points of operators* $\mathsf{Der}_{Br,ar}$ *exist.*

Proof. Since the rules of the constructors are identical, the proof of the first statement is entirely analogous to that of the intensional level of the Minimalist Foundation. Then, it is enough to prove the second statement for coinductive predicates. If the theory admits a greatest fixed point for every endofunctor of the form $\mathsf{Der}_{Br,ar}$, then, by Proposition 2.5.7, it equivalently admits one also for every endofunctor of the form $\mathsf{Conf}_{I,C}$; the former can then be used to interpret the coinductive predicates $\mathsf{CoInd}_{I,C}(x)$. □

Since the hypotheses of the above proposition only require the existence of a greatest fixed point, it follows as an immediate corollary that any extension of $\mathbf{ML}_1^\eta$ in which dependent M-types exist encodes coinductive predicates.

Finally, notice that if we were to take the alternative route of semantically defining coinductive predicates as terminal coalgebras of the endofunctors $\mathsf{Conf}_{I,C}$, Proposition 2.5.7 would have implied that they are a subclass of M-types.

2.6 Compatibility results

The results obtained in the previous sections allow us to partially answer the question posed in [Mai19] by extending some of the compatibility results of Section 1.8 to (co)inductive definitions.

Corollary 2.6.1.

1. $\mathbf{ML}_1^\eta + \mathsf{W}$ *is a compatible extension of* $\mathbf{mTT}_{\mathsf{ind}}$*;*

2. $\mathbf{mTT}_{\mathsf{ind}}$ *is compatible with* $\mathbf{HoTT}$*;*

3. $\mathbf{ML}_1^\eta + \mathsf{W} + \mathsf{funext} + \mathsf{K}$ *is a compatible extension of* $\mathbf{mTT}_{\mathsf{cind}}$*;*

4. $\mathbf{CC}_{\mathsf{ML}}$ *is a compatible extension of* $\mathbf{mTT}_{\mathsf{cind}}$*;*

5. $\mathcal{T}_{\mathbf{ArithTopos}}$ *is a compatible extension of* $\mathbf{emTT}_{\mathsf{cind}}$*.*

6. $\mathbf{emTT}_{\mathsf{ind}}$ *is compatible with* $\mathbf{CZF} + \mathsf{REA}$*;*

7. $\mathbf{emTT}_{\mathsf{cind}}$ *is compatible with* $\mathbf{CZF} + \mathsf{RRS}\text{-}\bigcup\mathsf{REA}$*.*

In the last two points above, REA *and* $\mathsf{RRS}\text{-}\bigcup\mathsf{REA}$ *are the Extension Axiom schemes defined for Regular sets and Strongly Regular sets satisfying the Relation Reflection Scheme, respectively. They were introduced to accommodate inductive and coinductive definitions in constructive set theory; for their precise statements see [AR10].*

Proof. 1. Inductive predicates can be interpreted straightforwardly into their analogues defined in Martin-Löf's type theory. In turn, thanks to Theorem 2.5.2, we know how to construct the latter using W-types.

2. Despite $\mathbf{HoTT}$ being an extension of $\mathbf{ML}_1^\eta + \mathsf{W}$, the interpretation defined in the previous point ceases to be compatible once the target theory is changed to $\mathbf{HoTT}$. This is because propositions of the Minimalist Foundation should be interpreted as h-propositions. Moreover, as already observed in the case of inductive basic covers in [CT20], the propositional truncation of a dependent W-type does not produce an inductive predicate; the solution adopted there, which also works for interpreting inductive predicates, is to make more use of the expressive power of Higher Inductive Types which allow postulating an introduction constructor ind *at the same time* with another constructor whose function is to trivialise the identity type.

3. It is enough to extend the interpretation described in the first point by interpreting $\mathbf{mTT}$-coinductive predicates as $\mathbf{ML}_1^\eta$-coinductive predicates, once we know that the target theory supports them since by the results in [AGMM15] it satisfies the hypotheses of Proposition 2.5.8.

4. By Proposition 2.3.1.
5. By Remark 2.2.5.
6. By Theorem 4.6 of [MMR22].
7. By Theorem 13.2.3 of [AR10].

□

Remark 2.6.2. In [MMR22], the authors proved a compatibility result for coinductive predicates in Martin-Löf's type theory alternative to the one in the second point of the above Corollary. There, instead of constructing coinductive predicates as M-types, they assumed in the target theory the existence of a Palmgren's superuniverse and encoded them directly.

Remark 2.6.3. The compatibility of $\mathbf{mTT}_{\mathsf{cind}}$ with **HoTT** remains an open problem. This is because, to adapt the interpretation described in [CM22] to account for coinductive predicates, one should be able to coinductively generate h-propositions in **HoTT**, but it is not evident how and if this can be done.

Chapter 3

Reversing the level structure

Chapter Abstract

In this chapter, we reverse the level structure of the Minimalist Foundation by defining an interpretation of the intensional level into the extensional one using in particular the technique of canonical isomorphisms (see [Mai09; Hof95]). Together with the quotient interpretation of [Mai09], we will deduce the equiconsistency of the two levels of **MF** and their equivalence over second-order arithmetic.

This chapter is adapted from joint work with M. E. Maietti [MS24].

3.1 Overview

The presence of an intensional and an extensional level in the Minimalist Foundation resembles very closely the two versions, intensional and extensional, of Martin-Löf's type theory. Indeed, both **emTT** and **mTT** are formulated as dependent type theories extending versions of Martin-Löf's type theory enriched with a primitive notion of proposition. More precisely, **emTT** extends the extensional version in [Mar84], while **mTT** extends the intensional one in [NPS90].

While it is notoriously difficult to interpret the extensional version of Martin-Löf's type theory in [Mar84] into its intensional one in [NPS90], especially in the presence of universes (see for example [Hof95; Pal22]), in the other direction the task is trivial. Indeed, the extensional version is a direct extension of the intensional one obtained mainly by strengthening the elimination rule of the identity type to make it reflect judgemental equality.

In the case of the Minimalist Foundation, the two levels, namely **mTT** and **emTT**, are conceptually linked according to Sambin's *forget-restore* design principle [SV98], stating that extensional concepts should be obtained by abstraction of the intensional ones in such a way that the process can be reverted at will; see Section 1.4 for more details. This principle was implemented as an interpretation of **emTT** in a quotient model of so-called setoids constructed over **mTT** in [Mai09]. However, contrary to Martin-Löf's type theory, **mTT** is not a direct extension of **emTT**, as we will show next. The question of whether the intensional level **mTT** can be interpreted into the extensional one **emTT** is therefore not trivial, and it is what we are going to answer positively in this chapter. Together with the quotient interpretations, this will grant important corollaries: the equiconsistency of the two levels, and the fact that they prove the exact same sentences expressible in the language of second-order arithmetic.

Our goal will be reached using a bridge theory between **mTT** and **emTT**, namely the extension **emTT** + propext of **emTT** obtained by adding axioms for *propositional extensionality* to it. On the one hand, it will be easy to show that **emTT**+propext is an extension of **mTT**; on the other, we will show that **emTT** + propext is interpretable back in **emTT** by employing the technique of *canonical isomorphisms*, already used in the interpretation of **emTT** in **mTT** in [Mai09], independently adopted for interpretations in other type-theoretic systems in [Hof95; Spa23], and later employed also in [CM22] to show the compatibility of **emTT** with **HoTT**. As as a byproduct, we will also conclude that **emTT** + propext is a *conservative* extension of **emTT**.

3.2 Propositional extensionality

The only technical obstacle preventing us from seeing **emTT** as a direct extension of **mTT** is the discrepancy between the extensional collection of small propositions up to equiprovability $\mathcal{P}(1)$, and the intensional universe $\mathsf{Prop_s}$ of small propositions; in particular, it is clear that $\mathcal{P}(1)$ cannot interpret $\mathsf{Prop_s}$ since the derived rule of the former on the left is weaker than the computation rule of the latter on the right:

$$\frac{\varphi\ prop_s}{\mathsf{Dc}([\varphi]) \Leftrightarrow \varphi\ true} \qquad\qquad \frac{\varphi\ prop_s}{\mathsf{T}(\widehat{\varphi}) = \varphi\ prop_s}$$

To rectify this situation, we consider adding to **emTT** axioms for propositional extensionality (for our purposes, only the second of the following two rules would suffice; we will include both for symmetry).

$$\mathsf{propext}\ \frac{\varphi\ prop \quad \psi\ prop \quad \varphi \Leftrightarrow \psi\ true}{\varphi = \psi\ prop}$$

$$\mathsf{prop_sext}\ \frac{\varphi\ prop_s \quad \psi\ prop_s \quad \varphi \Leftrightarrow \psi\ true}{\varphi = \psi\ prop_s}$$

Identifying equality and equiprovability for propositions clearly fixes the discrepancy between the two universes. The resulting theory will be called **emTT** + propext. As the next proposition shows, it is indeed an extension of **mTT** (besides one of **emTT**), obtained by strengthening judgemental equality and adding the quotient set constructor.

Proposition 3.2.1. **emTT** + propext *is an extension of* **mTT** *up to equivalence of theories.*

Proof. To show that **emTT** + propext can be obtained – up to renaming – as an extension by rules of **mTT**, one first makes the following renaming in the pre-syntax

$$\begin{aligned} \mathsf{Id} &\mapsto \mathsf{Eq} \\ \mathsf{Prop_s} &\mapsto \mathcal{P}(1) \\ \widehat{-} &\mapsto [-] \\ \mathsf{T}(-) &\mapsto \mathsf{Dc}(-) \end{aligned}$$

and then, adds all the rules of **emTT** + propext that are not already present in **mTT**, namely:

1. the rules of the quotient set constructor;
2. the η-conversion rule for dependent products ηC-Π;
3. the rules for proof-irrelevance prop-mono and prop-true; in particular, the canonical proof-term true of **emTT** can then interpret all the intensional proof-term constructors of **mTT**;
4. the equality reflection rule E-Eq;
5. congruence rules for types and terms constructors;
6. structural rules for embedding equalities;
7. the rule
$$\mathsf{F_s\text{-}Eq\text{-}}\mathcal{P}(1)\ \frac{U \in \mathcal{P}(1) \quad V \in \mathcal{P}(1)}{\mathsf{Eq}(\mathcal{P}(1), U, V)\ prop_s}$$
8. the two axioms of propositional extensionality $\mathsf{propext_s}$ and propext.

The theory obtained is exactly **emTT** + propext, because the rules of **mTT** that are not postulated in **emTT** + propext are nevertheless derivable in there, most notably the computation rule $\mathsf{C\text{-}Prop_s}$ thanks to the propositional extensionality axiom. □

Remark 3.2.2. It is easy to see that the quotient model used to interpret **emTT** in **mTT** actually interprets also **emTT** + propext. Indeed, the additional axioms of propositional extensionality are validated in the quotient model since equality of **emTT**-collections are interpreted as the existence of a so-called *canonical isomorphism* between the **mTT**-*extensional collections* interpreting them; however, in the case of extensional collections interpreting **emTT**-propositions, the existence of a canonical isomorphism amounts exactly to their equiprovability.

The main result of this chapter shows that **emTT** + propext can be interpreted back in **emTT**. The key idea is to interpret a proposition of **emTT** + propext as a proposition of **emTT** *up to equiprovability*, that is as an equivalence class of logically equivalent **emTT**-propositions. Since collections may depend on propositions, crucially thanks to the prop-into-col rule and the quotient set constructor, we will have to extend this rationale to all types of **emTT** by interpreting them as types *up to equivalent logical components*. The formalisation of this idea is achieved through the notion of canonical isomorphisms.

3.3 Canonical Isomorphisms

Since in this subsection our only object theory will be **emTT**, we assume that all the judgements are meant to be judgements derivable in **emTT**.

Definition 3.3.1 (Canonical isomorphisms)**.** We inductively define a family of functional terms, called *canonical isomorphisms*, between dependent collections. Both the dependent collections and the canonical isomorphisms between them will always be considered up to judgemental equality.

As customary, the common context is left implicit in each of the following clauses.

1. if φ and ψ are logically equivalent propositions (that is, if $\varphi \Leftrightarrow \psi$ *true* is derivable), then the unique functional term $\mathsf{true} \in \psi\ [x \in \varphi]$ is a canonical isomorphism;

2. the identities of the base types N_0, N_1, and $\mathcal{P}(1)$ are canonical isomorphisms;

3. if $\tau(x) \in B$ $[x \in A]$ is a canonical isomorphism between dependent sets, then the functional term

$$(a_1, \ldots, a_n) \in \mathsf{List}(A) \mapsto (\tau(a_1), \ldots, \tau(a_n)) \in \mathsf{List}(B)$$

extending $\tau(x)$ to lists element-wise is a canonical isomorphism; it can be formally defined as

$$\mathsf{El}_{\mathsf{List}}(l, \epsilon, (x, y, z).\mathsf{cons}(z, \tau(x))) \in \mathsf{List}(B)\ [l \in \mathsf{List}(A)]$$

4. if $\tau(x) \in A'$ $[x \in A]$ and $\sigma(x) \in B'$ $[x \in B]$ are two canonical isomorphisms between dependent sets, then their coproduct

$$\mathsf{inl}(a) \in A + B \mapsto \mathsf{inl}(\tau(a)) \in A' + B'$$
$$\mathsf{inr}(b) \in A + B \mapsto \mathsf{inr}(\sigma(b)) \in A' + B'$$

is a canonical isomorphism; it can be formally defined as

$$\mathsf{El}_{+}(z, (x).\tau(x), (y).\sigma(y)) \in A' + B'\ [z \in A + B]$$

5. if $B(x)$ *col* $[x \in A]$ and $B'(x)$ *col* $[x \in A']$ are two dependent collections, and there are canonical isomorphisms

$$\tau(x) \in A'\ [x \in A] \qquad \sigma(x, y) \in B'(\tau(x))\ [x \in A, y \in B(x)]$$

then the functional term

$$\langle a, b \rangle \in (\Sigma x \in A)B(x) \mapsto \langle \tau(a), \sigma(a, b) \rangle \in (\Sigma x \in A')B'(x)$$

is a canonical isomorphism; it can be formally defined as

$$\mathsf{El}_{\Sigma}(z, (x, y).\langle \tau(x), \sigma(x, y) \rangle) \in (\Sigma x \in A')B'(x)\ [z \in (\Sigma x \in A)B(x)]$$

6. let $B(x)$ *col* $[x \in A]$ and $B'(x)$ *col* $[x \in A']$ be two dependent collections such that their dependent products are well-formed collections – that is, either A, A', B, and B' are all sets, or A and A' are sets and B and B' are the constant family $\mathcal{P}(1)$; if there are canonical isomorphisms

$$\tau(x) \in A\ [x \in A'] \qquad \sigma(x, y) \in B'(x)\ [x \in A', y \in B(\tau(x))]$$

then the following is a canonical isomorphism

$$(\lambda x \in A')\sigma(x, \mathsf{Ap}(f, \tau(x))) \in (\Pi x \in A')B'(x)\ [f \in (\Pi x \in A)B(x)]$$

7. if $\tau(x) \in B\ [x \in A]$ is a canonical isomorphism between sets, $R(x,y)$ is a small equivalence relation on A, and $S(x,y)$ is a small equivalence relation on B such that $R(x,y) \Leftrightarrow S(\tau(x),\tau(y))\ \mathit{true}\ [x,y \in A]$ holds, then the functional term

$$[a] \in A/R \mapsto [\tau(a)] \in B/S$$

obtained by passing $\tau(x)$ to the quotient is a canonical isomorphism; it can be formally defined as

$$\mathsf{El}_{\mathsf{Q}}(z,(x).[\tau(x)]) \in B/S\ [z \in A/R]$$

We now derive some fundamental properties about canonical isomorphisms.

Lemma 3.3.2. *If* $\tau \in B\ [\Gamma, x \in A]$ *is a canonical isomorphism and* $\gamma \in \Gamma\ [\Delta]$ *is a telescopic substitution, then also* $\tau[\gamma,x] \in B[\gamma]\ [\Delta, x \in A[\gamma]]$ *is a canonical isomorphism.*

Proof. By induction on the definition of canonical isomorphism; in particular, for the case of propositions, by the fact that judgement derivability is closed under substitution, so that $\varphi \Leftrightarrow \psi\ \mathit{true}$ implies $\varphi[\gamma] \Leftrightarrow \psi[\gamma]\ \mathit{true}$.

We spell out the case of dependent product. Our goal is to show that

$$((\lambda x \in A')\sigma[\mathsf{Ap}(f,\tau)/y])[\gamma] \in ((\Pi x \in A')B')[\gamma]\ [\ \Delta, f \in ((\Pi x \in A)B)[\gamma]\ ]$$

is a canonical isomorphism. By IH we know that $\tau[\gamma,x] \in A[\gamma]\ [\Delta, x \in A'[\gamma]]$ is a canonical isomorphism. Now consider the extended telescopic substitution $\gamma, x \in (\Gamma, x \in A')\ [\Delta, x \in A'[\gamma]]$, and notice that $B[\tau/x][\gamma,x] \equiv B[\gamma,\tau[\gamma]]$; again by IH, we obtain that the following is a canonical isomorphism.

$$\sigma[\gamma,x,y] \in B'[\gamma,x]\ [\ \Delta, x \in A'[\gamma], y \in B[\gamma,\tau[\gamma]]\ ]$$

Observe that we have the following syntactical equalities.

$$((\lambda x \in A')\sigma[\mathsf{Ap}(f,t)/y])[\gamma] \equiv$$
$$(\lambda x \in A'[\gamma])\sigma[\gamma,\mathsf{Ap}(f,\tau[\gamma])] \equiv$$
$$(\lambda x \in A'[\gamma])\sigma[\gamma,x,y][\mathsf{Ap}(f,\tau[\gamma])/y]$$

By definition we know that the last term is a canonical isomorphism. □

Proposition 3.3.3. *Canonical isomorphisms enjoy the following properties:*

1. *identities are canonical isomorphisms;*
2. *canonical isomorphisms are indeed isomorphisms, and their inverses are again canonical isomorphisms;*
3. *the composition of two (composable) canonical isomorphisms is a canonical isomorphism;*
4. *there exists at most one canonical isomorphism between each pair of collections.*

Proof. The proof is analogous to the one performed for **HoTT** in Proposition 4.11 of [CM22].

Point 1 follows by induction on the construction of the collection, exploiting the η-equalities of the corresponding constructors.

Point 2 follows by induction on the definition of canonical isomorphism; in particular, thanks to the fact that $\Leftrightarrow$ is symmetric in the case of propositions. We spell out the case of dependent products. By induction hypothesis, there exist the two canonical inverses

$$\tau^{-1}(x) \in A' \; [x \in A] \qquad \sigma^{-1}(x, y) \in B(\tau(x)) \; [x \in A', y \in B'(x)]$$

if we substitute the second by the first we obtain

$$\sigma^{-1}(\tau^{-1}(x), y) \in B(\tau(\tau^{-1}(x))) = B(x) \; [x \in A, y \in B'(\tau^{-1}(x))]$$

which is again canonical thanks to Lemma 3.3.2, so that we can consider the canonical isomorphism

$$(\lambda x \in A)\sigma^{-1}(\tau^{-1}(x), f(\tau^{-1}(x))) \in (\Pi x \in A)B(x) \; [f \in (\Pi x \in A')B'(x)]$$

which can be easily checked to be the inverse.

For *point 3*, observe that two objects are related by a canonical isomorphism only if they have the same outermost type constructor or if they are both propositions; in the latter case, we rely on the transitivity of $\Leftrightarrow$; in the former case, we proceed by induction on the outermost type constructor. We spell out the case of the dependent product. Suppose to have the following canonical isomorphisms

$$\begin{aligned}
&\tau(x) \in A\ [x \in A'] \\
&\tau'(x) \in A'\ [x \in A''] \\
&\sigma(x, y) \in B'(x)\ [x \in A', y \in B(\tau(x))] \\
&\sigma'(x, y) \in B''(x)\ [x \in A'', y \in B'(\tau'(x))]
\end{aligned}$$

By Lemma 3.3.2 also the following substituted morphism is canonical

$$\sigma(\tau'(x), y) \in B'(\tau'(x))\ \ [x \in A'', y \in B(\tau(\tau'(x)))]$$

By inductive hypothesis, the following isomorphisms obtained by composition are canonical

$$\tau(\tau'(x)) \in A\ [x \in A''] \quad \sigma'(x, \sigma(\tau'(x), y)) \in B''(x)\ [x \in A'', y \in B(\tau(\tau'(x)))]$$

We must check that the composition of the two canonical isomorphisms

$$\begin{gathered}
(\lambda x \in A')\sigma(x, \mathsf{Ap}(f, \tau(x))) \in (\Pi x \in A')B'(x)\ [f \in (\Pi x \in A)B(x)] \\
(\lambda x \in A')\sigma'(x, \mathsf{Ap}(f, \tau'(x))) \in (\Pi x \in A'')B''(x)\ [f \in (\Pi x \in A')B'(x)]
\end{gathered}$$

is canonical, but their composition is equal to

$$\begin{aligned}
&(\lambda x \in A'')\sigma'(x, \mathsf{Ap}((\lambda x \in A')\sigma(x, \mathsf{Ap}(f, \tau(x))), \tau'(x))) = \\
&(\lambda x \in A'')\sigma'(x, \sigma(\tau'(x), \mathsf{Ap}(f, \tau(\tau'(x)))))
\end{aligned}$$

which is canonical by definition.

For *point 4*, recall that canonical isomorphisms are considered up to judgemental equality. Then, the statement is trivial in the case of propositions; in the other cases, it is proven by induction on the outermost type constructor of the two collections. □

We can extend the notion of canonical isomorphisms to contexts of **emTT**.

Definition 3.3.4. We inductively define a family of telescopic substitutions between contexts, called again *canonical isomorphisms*. Again, we will consider them up to judgemental equality.

- the empty telescopic substitution between empty contexts $() \in ()\ [\,]$ is a canonical isomorphism;

- if A *col* $[\Gamma]$ and B *col* $[\Delta]$ are two dependent collections, $\sigma \in \Delta\ [\Gamma]$ is a canonical isomorphism between contexts, and $\tau \in B[\sigma]\ [\Gamma, x \in A]$ is a canonical isomorphism between collections, then the extension $\sigma, \tau \in (\Delta, x \in B)\ [\Gamma, x \in A]$ is a canonical isomorphism.

It is easy to check by induction that canonical isomorphisms between contexts inherit the properties of Proposition 3.3.3.

Definition 3.3.5. We say that *two contexts* Γ *and* Δ *are canonically isomorphic* if there exists a (necessarily unique) canonical isomorphism between them.

We say that *two dependent types* A *type* $[\Gamma]$ *and* B *type* $[\Delta]$ *are canonically isomorphic* if their extended contexts $\Gamma, x \in A$ and $\Delta, y \in B$ are canonically isomorphic; equivalently, if their contexts Γ and Δ are canonically isomorphic, and there exists a (necessarily unique) canonical isomorphism between A and $B[\sigma]$, where $\sigma \in \Delta\ [\Gamma]$ is the canonical isomorphism between contexts.

Finally, we say that *two telescopic substitutions* $\gamma \in \Gamma\ [\Gamma']$ *and* $\delta \in \Delta\ [\Delta']$ *are canonically isomorphic* if both their domain and codomain are canonically isomorphic and the compositions of telescopic substitutions depicted pictorially in the following square are judgmentally equal

$$\begin{array}{ccc} \Gamma' & \xrightarrow{\gamma} & \Gamma \\ {\scriptstyle\sigma'}\downarrow & & \downarrow{\scriptstyle\sigma} \\ \Delta' & \xrightarrow{\delta} & \Delta \end{array}$$

where σ and σ' are the canonical isomorphisms between contexts. As a special case of the latter definition, we say that two dependent terms are canonically isomorphic if they are so as telescopic substitutions; namely, *two terms* $a \in A\ [\Gamma]$ *and* $b \in B\ [\Delta]$ *are canonically isomorphic* if the dependent collections they belong to are canonically isomorphic and the following equality holds

$$\tau(a) = b[\sigma] \in B[\sigma]\ [\Gamma]$$

where $\sigma \in \Delta\ [\Gamma]$ is the canonical isomorphism between contexts, and $\tau(x) \in B[\sigma]\ [\Gamma, x \in A]$ is the canonical isomorphism between collections.

Remark 3.3.6. We could have organised the definitions of canonical isomorphisms using the language of category theory. In particular, we could have considered the syntactic category **Ctx** of contexts and telescopic substitutions up to judgemental equality. In that case, the square depicted above in the definition of canonically isomorphic contexts could have been interpreted as a diagram of **Ctx**, and formally required to be commutative.

The first three points of Proposition 3.3.3 imply that being canonically isomorphic (for contexts, collections, telescopic substitutions and terms) is an equivalence relation. Moreover, the following property of preservation under substitution holds.

Lemma 3.3.7. *Let $\gamma \in \Gamma\ [\Gamma']$ and $\delta \in \Delta\ [\Delta']$ be two canonically isomorphic telescopic substitutions. If A type $[\Gamma]$ and B type $[\Delta]$ are canonically isomorphic types, then also $A[\gamma]$ type $[\Gamma']$ and $B[\delta]$ type $[\Delta']$ are canonically isomorphic types. Moreover, if $a \in A\ [\Gamma]$ and $b \in B\ [\Delta]$ are canonically isomorphic terms, then also $a[\gamma] \in A[\gamma]\ [\Gamma']$ and $b[\delta] \in B[\delta]\ [\Delta']$ are canonically isomorphic terms.*

Proof. Assume $\sigma \in \Delta\ [\Gamma]$ and $\tau \in B[\sigma]\ [\Gamma, x \in A]$ are the canonical isomorphisms induced by the hypotheses. Then, by Lemma 3.3.2, we know that also $\tau[\gamma, x] \in B[\sigma][\gamma]\ [\Gamma', x \in A[\gamma]]$ is canonical, but $B[\sigma][\gamma] \equiv B[\delta][\sigma']$ by definition of canonically isomorphic generalised substitutions, thus the substituted types are canonically isomorphic. For terms, we know by hypothesis that $\tau(a) = b[\sigma]$; thus, by closure of judgements under substitutions we conclude

$$\tau[\gamma, a[\gamma]] = b[\sigma][\gamma] \equiv b[\delta][\sigma]$$

□

Finally, we notice that we can always correct a type (resp. a term) into a canonically isomorphic one to match a given context (type) canonically equivalent to the original one.

Lemma 3.3.8. *Let A type $[\Gamma]$, and Δ ctx canonically isomorphic to Γ ctx, then there exists $\tilde{A}$ type $[\Delta]$ canonically isomorphic to A type $[\Gamma]$.*

Analogously, if $a \in A\ [\Gamma]$ is a term and B col $[\Delta]$ is a collection canonically isomorphic to A col $[\Gamma]$, then there exists a term $\tilde{a} \in B\ [\Delta]$ canonically isomorphic to $a \in A\ [\Gamma]$.

Proof. Consider $\tilde{A} :\equiv A[\sigma^{-1}]$ *type* $[\Delta]$ and $\tilde{a} :\equiv \tau(a)[\sigma^{-1}] \in B\ [\Delta]$, where $\sigma \in \Delta\ [\Gamma]$ and $\tau \in B[\sigma]\ [\Gamma, x \in A]$ are some assumed existing canonical isomorphisms. The same σ and τ witness that A *type* $[\Gamma]$ and $\tilde{A}$ *type* $[\Delta]$ are canonically isomorphic types, and that $a \in A$ $[\Gamma]$ and $\tilde{a} \in B$ $[\Delta]$ are canonically isomorphic terms □

3.4 Conservativity of propositional extensionality

With the machinery of canonical isomorphisms set up, we are ready to interpret **emTT** + propext into **emTT**. The idea is to define an identity interpretation *up to canonical isomorphisms*.

As customary in type theory, we first define *a priori partial* interpretation functions on the pre-syntax of **emTT** + propext; the Validity Theorem 3.4.3 will ensure that such functions are total when restricted to the derivable judgements of **emTT** + propext. More in detail, we define three partial functions which send:

1. context judgements Γ *ctx* to an equivalence class $[\![\, \Gamma\ ctx \,]\!]$ of canonically isomorphic **emTT**-contexts;
2. type judgements A *type* $[\Gamma]$ to an equivalence class $[\![\, A\ type\ [\Gamma] \,]\!]$ of canonically isomorphic **emTT**-collections such that all its representatives are defined in contexts belonging to $[\![\, \Gamma\ ctx \,]\!]$, and such that at least one among them is of kind *type*;
3. term judgements $a \in A$ $[\Gamma]$ to an equivalence class $[\![\, a \in A\ [\Gamma] \,]\!]$ of canonically isomorphic **emTT**-terms such that all its representatives are defined in contexts belonging to $[\![\, \Gamma\ ctx \,]\!]$.

In the following we use the upper corner notation $\ulcorner - \urcorner$ to denote equivalence classes of canonically isomorphic expressions.

Definition 3.4.1 (Interpretation)**.** The three functions specified above are defined by recursion on the pre-syntax of **emTT** + propext where, in each clause, we interpret the constructor in case (be it of contexts, types or terms) with the same constructor in the target theory **emTT**. We spell out the case of contexts, variables, the canonical true term, the existential quantifier, and dependent product.

Contexts and variables.

- $[\![\, () \; ctx \,]\!] :\equiv \ulcorner () \; ctx \urcorner$
- $[\![\, \Gamma, x \in A \; ctx \,]\!] :\equiv \ulcorner \Gamma', x \in A' \; ctx \urcorner$
  provided that $[\![\, A \; col \; [\Gamma] \,]\!] \equiv \ulcorner A' \; col \; [\Gamma'] \urcorner$
- $[\![\, x \in A \; [\Gamma, x \in A, \Delta] \,]\!] :\equiv \ulcorner x \in A' \; [\Gamma', x \in A', \Delta'] \urcorner$
  provided that $[\![\, \Gamma, x \in A, \Delta \; ctx \,]\!] \equiv \ulcorner \Gamma', x \in A', \Delta' \; ctx \urcorner$

True term.

- $[\![\, \mathsf{true} \in \varphi \; [\Gamma] \,]\!] :\equiv \ulcorner \mathsf{true} \in \varphi' \; [\Gamma'] \urcorner$
  provided that $[\![\, \varphi \; prop \; [\Gamma] \,]\!] :\equiv \ulcorner \varphi' \; prop \; [\Gamma'] \urcorner$

Existential quantifier.

- $[\![\, (\exists x \in A)\varphi \,|\, \Gamma \,]\!] :\equiv \ulcorner (\exists x \in A')\varphi' \; prop \; [\Gamma'] \urcorner$
  provided that $[\![\, \varphi \,|\, \Gamma, x \in A \,]\!] \equiv \ulcorner \varphi' \; prop \; [\Gamma', x \in A'] \urcorner$

Dependent products.

- $[\![\, (\Pi x \in A)B \; type \; [\Gamma] \,]\!] :\equiv \ulcorner (\Pi x \in A')B' \; type \; [\Gamma'] \urcorner$
  provided that $[\![\, B \; type \; [\Gamma, x \in A] \,]\!] \equiv \ulcorner B' \; type \; [\Gamma', x \in A'] \urcorner$
- $[\![\, (\lambda x \in A)b \in (\Pi x \in A)B \; [\Gamma] \,]\!] :\equiv \ulcorner (\lambda x \in A')b' \in (\Pi x \in A')B' \; [\Gamma'] \urcorner$
  provided that $[\![\, b \in B \; [\Gamma, x \in A] \,]\!] \equiv \ulcorner b' \in B' \; [\Gamma', x \in A'] \urcorner$
- $[\![\, \mathsf{Ap}(f, a) \in B[a/x] \; [\Gamma] \,]\!] :\equiv \ulcorner \mathsf{Ap}(f', a') \in B'[a'/x] \; [\Gamma'] \urcorner$
  provided that $[\![\, f \in (\Pi x \in A)B \; [\Gamma] \,]\!] \equiv \ulcorner f' \in (\Pi x \in A')B' \; [\Gamma'] \urcorner$
  and $[\![\, a \in A \; [\Gamma] \,]\!] \equiv \ulcorner a' \in A' \; [\Gamma'] \urcorner$

The interpretation of the other constructors is defined analogously.

To smoothly state the substitution lemma, we define in an analogous way a fourth partial function sending judgements of the derived form $\gamma \in \Gamma \; [\Delta]$ to an equivalence class of **emTT**-canonically isomorphic telescopic substitutions $[\![\, \gamma \in \Gamma \; [\Delta] \,]\!]$ defined in contexts belonging to $[\![\, \Delta \; ctx \,]\!]$. We then have the following.

Lemma 3.4.2 (Substitution)**.** *Assume* $[\![\, \gamma \in \Gamma \; [\Delta] \,]\!] \equiv \ulcorner \gamma' \in \Gamma' \; [\Delta'] \urcorner$ *holds, then:*

1. $[\![A \ type\ [\Gamma]]\!] \equiv \ulcorner A' \ type\ [\Gamma'] \urcorner$ *implies* $[\![A[\gamma] \ type\ [\Delta]]\!] \equiv \ulcorner A'[\gamma'] \ type\ [\Delta'] \urcorner$;
2. $[\![a \in A\ [\Gamma]]\!] \equiv \ulcorner a' \in A'\ [\Gamma'] \urcorner$ *implies*
   $[\![a[\gamma] \in A[\gamma]\ [\Delta]]\!] \equiv \ulcorner a'[\gamma'] \in A'[\gamma']\ [\Delta'] \urcorner$.

Proof. By induction on the expressions A and a. □

Theorem 3.4.3 (Validity). *1. if* **emTT** + propext $\vdash \Gamma\ ctx$, *then* $[\![\Gamma]\!]$ *is defined;*

2. *if* **emTT** + propext $\vdash A\ type\ [\Gamma]$, *then* $[\![A\ type\ [\Gamma]]\!]$ *is defined;*
3. *if* **emTT** + propext $\vdash a \in A\ [\Gamma]$, *then* $[\![a \in A\ [\Gamma]]\!]$ *is defined and all its terms are defined in types belonging to* $[\![A\ col\ [\Gamma]]\!]$;
4. *if* **emTT**+propext $\vdash A = B\ type\ [\Gamma]$, *then* $[\![A\ type\ [\Gamma]]\!] \equiv [\![B\ type\ [\Gamma]]\!]$;
5. *if* **emTT** + propext $\vdash a = b \in A\ [\Gamma]$, *then* $[\![a \in A\ [\Gamma]]\!] \equiv [\![b \in A\ [\Gamma]]\!]$.

Proof. By induction on the derivations of **emTT**+propext, using Proposition 3.3.3 and Lemmas 3.3.7, 3.3.8, and 3.4.2. In most cases, it is trivial to check that the **emTT**-judgements used in the interpretation are actually derivable, and that the side condition on the contexts holds. Therefore, we mainly need to check that the definition of the equivalence classes does not depend on the choice of representatives, and that the rules of **emTT**+propext are validated. We spell out some of the most relevant cases.

Existential quantifier. Assume that $\varphi\ prop\ [\Gamma, x \in A]$ is canonically isomorphic to $\psi\ prop\ [\Delta, y \in B]$; we want to show that their existential quantification are canonically isomorphic. By hypothesis, we have canonical isomorphisms

$$\sigma \in \Gamma\ [\Delta] \qquad \tau(x) \in B[\sigma]\ [\Gamma, x \in A]$$

and we know that $\varphi \Leftrightarrow \psi[\sigma, \tau]$ holds. The following chain of equiprovable propositions shows that also the existential propositions are equiprovable, and thus canonically isomorphic

$$\begin{aligned}(\exists x \in A)\varphi &\Leftrightarrow (\exists x \in A)\psi[\sigma, \tau] \\ &\Leftrightarrow (\exists y \in B[\sigma])\psi[\sigma, y] \\ &\equiv ((\exists y \in B)\psi)[\sigma]\end{aligned}$$

where in the first step we used the fact that existential quantification preserves equiprovability, and in the second one that reindexing of existential quantifiers along isomorphisms preserves equiprovability.

Lambda abstraction. Assume that $b \in B\ [\Gamma, x \in A]$ and $b' \in B'\ [\Gamma', x \in A']$ are two canonically isomorphic terms; we want to show that their lambda abstraction are again canonically isomorphic. By hypothesis, we know there are canonical isomorphisms

$$\sigma \in \Gamma'\ [\Gamma]$$
$$\tau(x) \in A'[\sigma]\ [\Gamma, x \in A]$$
$$\rho(x, y) \in B'[\sigma, \tau]\ [\Gamma, x \in A, y \in B]$$

such that

$$\rho(x, b) = b'[\sigma, \tau] \in B[\sigma, \tau]\ [\Gamma, x \in A] \tag{3.1}$$

By Proposition 3.3.3 and Lemma 3.3.2 also the following are canonical isomorphisms.

$$\tau^{-1}(x) \in A\ [\Gamma, x \in A'[\sigma]]$$
$$\rho(\tau^{-1}(x), y) \in B'[\sigma, x]\ [\Gamma, x \in A'[\sigma], y \in B(\tau^{-1}(x))]$$

Moreover, by applying the term ρ^{-1} to (1) also the following hold.

$$b = \rho^{-1}(x, b'[\sigma, \tau]) \in B\ [\Gamma, x \in A]$$
$$(\lambda x \in A)b = (\lambda x \in A)\rho^{-1}(x, b'[\sigma, \tau]) \in (\Pi x \in A)B\ [\Gamma]$$

By definition of canonical isomorphism between dependent products, we have that the term

$$\zeta(f) :\equiv (\lambda x \in A'[\sigma])s(\tau^{-1}(x), \mathsf{Ap}(f, \tau^{-1}(x)))$$

is a canonical isomorphisms between $(\Pi x \in A)B$ and $(\Pi x \in A'[\sigma])B'[\sigma, x] \equiv ((\Pi x \in A')B')[\sigma]$. Finally, we can check that

$$\begin{aligned}
\zeta((\lambda x \in A)b) &= \zeta((\lambda x \in A)\rho^{-1}(x, b'[\sigma, \tau])) \\
&= (\lambda x \in A'[\sigma])\rho(\tau^{-1}(x), \rho^{-1}(\tau^{-1}(x), b'[\sigma, \tau][\tau^{-1}/x])) \\
&= (\lambda x \in A'[\sigma])b'[\sigma, x] \\
&\equiv ((\lambda x \in A')b')[\sigma] \in ((\Pi x \in A')B')[\sigma]\ [\Gamma]
\end{aligned}$$

Thus, we have concluded that $(\lambda x \in A)b \in (\Pi x \in A)B\ [\Gamma]$ and $(\lambda x \in A')b' \in (\Pi x \in A')B'\ [\Gamma']$ are canonically isomorphic terms.

Propositional extensionality. By inductive hypothesis on the first premise, we know that, for some $\varphi'\ prop\ [\Gamma']$, we have $[\![\, \varphi\ prop\ [\Gamma]\,]\!] \equiv \ulcorner \varphi'\ prop\ [\Gamma'] \urcorner$; by inductive hypothesis on the second premise corrected by Lemma 3.3.8, we know that $[\![\, \psi\ prop\ [\Gamma]\,]\!] \equiv \ulcorner \psi'\ prop\ [\Gamma'] \urcorner$ for some proposition ψ' defined in the same context Γ' of φ'. By definition of the interpretation we then have $[\![\, \varphi \Leftrightarrow \psi\ prop\ [\Gamma]\,]\!] \equiv \ulcorner \varphi' \Leftrightarrow \psi'\ prop\ [\Gamma'] \urcorner$. Finally, by inductive hypothesis on the third premise, we know that the interpretation of $[\![\, \mathsf{true} \in \varphi \Leftrightarrow \psi\ prop\ [\Gamma]\,]\!]$ is well defined; in particular, this means that $\mathsf{true} \in \varphi' \Leftrightarrow \psi'$ is derivable in **emTT**, but this amounts to φ' and ψ' being canonically isomorphic, which in turn implies $[\![\, \varphi\ prop\ [\Gamma]\,]\!] \equiv \ulcorner \varphi'\ prop\ [\Gamma'] \urcorner \equiv \ulcorner \psi'\ prop\ [\Gamma'] \urcorner \equiv [\![\, \psi\ prop\ [\Gamma]\,]\!]$. □

Remark 3.4.4. As already mentioned, the idea of using canonical isomorphisms to interpret extensional equalities in type theory was originally conceived in [Mai09] between objects of a quotient model, and, independently by Hofmann in [Hof95], whilst with the additional help of the Axiom of Choice in the meta-theory. The results in [Hof95] were later made effective in [Our05; WST19] with the adoption of a heterogeneous equality.

Remark 3.4.5. The interpretation of **emTT** + propext into **emTT** defined above is *effective*, in the sense of [Mai09] and [WST19], since its Validity Theorem 3.4.3 can be constructively implemented as a translation of derivations of the source theory into derivations of the target theory. We leave such an implementation in a proof-assistant as future works.

The interpretation enjoys the following crucial (albeit trivial) property, which allows it to be seen as a retraction of the identity interpretation of **emTT** into **emTT** + propext.

Proposition 3.4.6. *1. If* **emTT** $\vdash \Gamma\ ctx$*, then* $[\![\, \Gamma\ ctx\,]\!] \equiv \ulcorner \Gamma\ ctx \urcorner$*;*

2. if **emTT** $\vdash A\ type\ [\Gamma]$*, then* $[\![\, A\ type\ [\Gamma]\,]\!] \equiv \ulcorner A\ type\ [\Gamma] \urcorner$*;*

3. if **emTT** $\vdash a \in A\ [\Gamma]$*, then* $[\![\, a \in A\ [\Gamma]\,]\!] \equiv \ulcorner a \in A\ [\Gamma] \urcorner$*.*

Proof. Straightforward by induction on the derivations of **emTT**. □

As an immediate consequence, we have that propositional extensionality is really a conservative assumption over **emTT**.

Corollary 3.4.7 (Conservativity)**.** *If* **emTT** $\vdash \varphi$ *prop* $[\Gamma]$ *and* **emTT** + $\mathsf{propext} \vdash \varphi$ *true* $[\Gamma]$*, then* **emTT** $\vdash \varphi$ *true* $[\Gamma]$*.*

Proof. By point 2 of Proposition 3.4.6, $[\![\, \varphi \; prop \; [\Gamma] \,]\!] \equiv \ulcorner \varphi \; prop \; [\Gamma] \urcorner$; then, by point 3 of Theorem 3.4.3, we know that $[\![\, \mathsf{true} \in \varphi \; [\Gamma] \,]\!]$ is defined and the types of its representatives are canonically isomorphic to φ *prop* $[\Gamma]$, which, therefore, is inhabited. □

After the above results, because **emTT** + propext seems to behave better than **emTT** with respect to **mTT**, and, at the same time, it is completely equivalent to **emTT** from a proof-theoretical point of view, it would be tempting to replace **emTT** with **emTT** + propext as the extensional level of the minimalist foundation. However, this cannot be done since adding propositional extensionality breaks the compatibility with the calculus $\mathcal{T}_{\mathbf{Topos}}$ for the internal language of toposes, where propext does not hold.

3.5 Equiconsistency of the two levels

Putting together the work of the previous sections we finally derive the following.

Corollary 3.5.1 (Equiconsistency)**.** *The theories* **mTT** *and* **emTT** *are equiconsistent.*

Proof. From the quotient interpretation of [Mai09], we know that the consistency of **mTT** implies that of **emTT**. For the other direction, we know that **emTT** + propext is an extension of **mTT** by Proposition 3.2.1; in turn, **emTT** + propext is equiconsistent to its fragment **emTT** by Corollary 3.4.7. □

Furthermore, with a little extra work, we can upgrade the equiconsistency result to a proof-theoretic equivalence over the language of second-order arithmetic.

Recall that the language of second-order arithmetic is two-sorted over predicate logic (where equality is not assumed as primitive). The sorts will be denoted as $\mathbb{N}$ and $\mathcal{P}(\mathbb{N})$. All the function symbols are relative to the sort

$\mathbb{N}$ and they are a constant $\mathsf{0}$ for the zero; a unary function S for the successor; and two binary functions $+$ and $\cdot$ for sum and multiplication, respectively. Its predicate symbols are a binary relations $n = m$ for equality between terms of sort $\mathbb{N}$; and a binary relation $n \,\varepsilon\, U$ for membership between a term n sorted by $\mathbb{N}$ and a term U sorted by $\mathcal{P}(\mathbb{N})$.

Such language admits an obvious interpretation both in **mTT** and **emTT**. In particular, in **mTT** the sort $\mathcal{P}(\mathbb{N})$ is interpreted as $\mathbb{N} \to \mathsf{Prop_s}$, the membership relation $n \,\varepsilon\, U$ as $\mathsf{T}(\mathsf{Ap}(U, n))$, and the equality $n = m$ as the intensional propositional equality $\mathsf{Id}(\mathbb{N}, n, m)$; while **emTT** interprets the sort $\mathcal{P}(\mathbb{N})$ as $\mathbb{N} \to \mathcal{P}(1)$, the equality relation $n = m$ as the extensional propositional equality $\mathsf{Eq}(\mathbb{N}, n, m)$, and the membership relation $n \,\varepsilon\, U$ as $\mathsf{Dc}(\mathsf{Ap}(U, n))$.

Corollary 3.5.2. *The theories* **mTT** *and* **emTT** *prove the same second-order arithmetic formulas.*

Proof. Let φ be a formula in the language of second-order arithmetic, and denote with $\varphi_{\mathbf{mTT}}$ and $\varphi_{\mathbf{emTT}}$ its interpretation as a proposition of **mTT** and **emTT**, respectively. Our goal is to show that $\mathbf{emTT} \vdash \varphi_{\mathbf{emTT}}\ true$ if and only if there exists a term p such that $\mathbf{mTT} \vdash p \in \varphi_{\mathbf{mTT}}$.

In one direction, observe that the embedding of **mTT** into **emTT** + $\mathsf{propext}$ in Proposition 3.2.1 translates $\varphi_{\mathbf{mTT}}$ into $\varphi_{\mathbf{emTT}}$, since the only differences in the interpretations are those regarding propositional equalities Id and Eq and the collections $\mathsf{Prop_s}$ and $\mathcal{P}(1)$. Now assume that $\mathbf{mTT} \vdash p \in \varphi_{\mathbf{mTT}}$ for some proof-term p; then, again by the embedding of Proposition 3.2.1, we obtain $\mathbf{emTT} + \mathsf{propext} \vdash \varphi_{\mathbf{emTT}}\ true$, and, by Corollary 3.4.7, also $\mathbf{emTT} \vdash \varphi_{\mathbf{emTT}}\ true$.

For the other direction, we use the restore interpretation of [Mai09], denoted as $(-)^{\mathcal{I}}$. Firstly, we show that the interpretation $\varphi^{\mathcal{I}}_{\mathbf{emTT}}$ of $\varphi_{\mathbf{emTT}}$ in **mTT** is equiprovable to $\varphi_{\mathbf{mTT}}$; more generally, we show that the quotient interpretation commutes, up to isomorphism, with the interpretations of second-order arithmetical sorts, terms, and formulas.

Logical symbols. The quotient interpretations fixes the falsum constant, the propositional connectives, and the quantifiers.

The sort $\mathbb{N}$ *and the equality relation.* Following the interpretation of the singleton set and the list constructor, it is easy to check that the extensional set $\mathbb{N}$ and its extensional propositional equality $\mathsf{Eq}(\mathbb{N}, x, y)$ can be interpreted in **mTT** as $\mathbb{N}$ and $\mathsf{Id}(\mathbb{N}, x, y)$, respectively.

Terms. Since the interpretation fixes the introduction and elimination terms for list, all the relevant natural number terms – namely, the zero constant, and the successor, addition, and multiplication operations – are fixed when passing from **emTT** to **mTT**.

The sort $\mathcal{P}(\mathbb{N})$*.* By the interpretation of the function space and the power-collection of the singleton, we have that the extensional collection $\mathcal{P}(\mathbb{N})$ is interpreted as the intensional collection

$$(\Sigma f \in \mathbb{N} \to \mathsf{Prop_s})(\forall x, y \in \mathbb{N})(\ \mathsf{Id}(\mathbb{N}, x, y) \Rightarrow (\mathsf{T}(f(x)) \Leftrightarrow \mathsf{T}(f(y)))\ )$$

which, in turn, is isomorphic to $\mathbb{N} \to \mathsf{Prop_s}$, since the proposition used to define it by comprehension is always satisfied.

The membership relation. Thanks to the interpretation of the propositional equality of the collection $\mathcal{P}(1)$, the interpretation of the relation of membership is interpreted as follows.

$$\begin{aligned}\mathsf{Dc}(\mathsf{Ap}(U, n))^{\mathcal{I}} &\equiv (\mathsf{Eq}(\mathcal{P}(1), \mathsf{Ap}(U, n), [\top]))^{\mathcal{I}} \\ &\equiv (\mathsf{T}(\mathsf{Ap}(U^{\mathcal{I}}, n^{\mathcal{I}})) \Leftrightarrow \top) \\ &\Leftrightarrow \mathsf{T}(\mathsf{Ap}(U^{\mathcal{I}}, n^{\mathcal{I}}))\end{aligned}$$

We can now conclude since, if $\mathbf{emTT} \vdash \varphi_{\mathbf{emTT}}$ *true*, then by the validity of the quotient interpretation there exists a proof-term p such that $\mathbf{mTT} \vdash p \in \varphi^{\mathcal{I}}_{\mathbf{emTT}}$, and thus, by equiprovability, also one for $\varphi_{\mathbf{mTT}}$. □

Remark 3.5.3. In the above corollary, it was crucial not to have assumed equality as a primitive logical symbol, but only as a relation between terms of $\mathbb{N}$. Its proof would not have worked if we had allowed formulas involving the equality of $\mathcal{P}(\mathbb{N})$ because the difference between intensionality and extensionality would have appeared.

Chapter 4

The classical version

Chapter Abstract

In this chapter, we extend to the Minimalist Foundation Gödel's double-negation of classical arithmetic into intuitionistic arithmetic. In this way, we show the equiconsistency of the Minimalist Foundation with its classical version, and its compatibility with classical predicativism à la Weyl. By adapting the same technique we show that also the Calculus of Constructions is equiconsistent with its classical version.

This chapter is adapted from joint work with M. E. Maietti [MS24].

4.1 Overview

In Chapter 3, we proved that the two levels of the Minimalist Foundation are equiconsistent (Corollary 3.5.1), and even equivalent when it comes to second-order arithmetic (Corollary 3.5.2). These results, together with the intended use of the two levels described in Section 1.4 justify the following definition.

Definition 4.1.1. We refer to the *classical counterpart* or *classical version* of **MF** as the theory $\mathbf{emTT}^c$ obtained by extending the extensional level **emTT** with the Law of Excluded Middle

$$\mathsf{LEM}\ \frac{\varphi\ prop}{\varphi \vee \neg\varphi\ true}$$

The main goal of this chapter is to show that $\mathbf{emTT}^c$ is equiconsistent with **emTT**. We will do so by adapting the Gödel-Gentzen double-negation translation of classical logic in the intuitionistic one (see for example [TD88]) to interpret $\mathbf{emTT}^c$ within **emTT**, exploiting in particular the fact that the type constructors of **emTT** preserve the $\neg\neg$-stability (from now on, just *stability*) of their propositional equalities.

As a consequence of the equiconsistency of **emTT** with $\mathbf{emTT}^c$, we will also show that real numbers à la Dedekind do not form a set neither in **emTT**, nor in $\mathbf{emTT}^c$. Therefore, as anticipated in Subsection 1.2.3, $\mathbf{emTT}^c$ can be taken as a foundation of classical predicative mathematics in the spirit of Weyl in [Wey18], and of course **MF** through **emTT** becomes compatible with it.

Finally, by exploiting the fact that the intensional level **mTT** is a *predicative version* of Coquand-Huet's Calculus of Constructions [CH88], we show that the chain of equiconsistency results for **MF** can be straightforwardly adapted to an impredicative version of **MF** whose intensional level is the Calculus of Constructions equipped with inductive types from the first-order fragment of **MLTT**, which we call $\mathbf{CC}_{\mathsf{ML}}$, thus extending the result in [Sel97] on the equiconsistency of the logical base of the calculus with its classical version without relying on normalisation properties of $\mathbf{CC}_{\mathsf{ML}}$.

4.2 The double-negation translation

In this section we extend the double-negation translation of predicate logic to account for the set-theory of the Minimalist Foundation in order to interpret the classical version $\mathbf{emTT}^c$ of its extensional level into the standard intuitionistic version **emTT**.

The underlying idea of the translation is straightforward: we want to keep translating propositions φ of $\mathbf{emTT}^c$ into stable propositions $\varphi^{\mathcal{N}}$ of **emTT** (i.e. those equivalent to their double negation) while leaving unaltered set-theoretical constructors that do not involve logic.

Formally, a proposition φ *prop* of **emTT** is said to be *stable* if the judgement $\neg\neg\varphi \Rightarrow \varphi$ *true* is derivable. Accordingly, a collection A *col* is said to have *stable equality* if its propositional equality $\mathsf{Eq}(A, x, y)$ *prop* $[x, y \in A]$ is stable in **emTT**.

Since **emTT** is a dependent type system in which sorts can depend on terms and propositions, the translation will be defined on all those entities, and not just on formulas.

Definition 4.2.1 (Interpretation)**.** We define by simultaneous recursion four endofunctions $(-)^{\mathcal{N}}$ on pre-contexts, pre-types, pre-propositions, and preterms, respectively.

Variables and contexts. The translation does not affect variables, and on contexts it is defined in the obvious way.

$$x^{\mathcal{N}} :\equiv x$$
$$()^{\mathcal{N}} :\equiv ()$$
$$(\Gamma, x \in A)^{\mathcal{N}} :\equiv \Gamma^{\mathcal{N}},\, x \in A^{\mathcal{N}}$$

Logic. We translate the falsum constant and the propositional connectives exactly as in the case of predicate logic.

$$
\begin{aligned}
\bot^{\mathcal{N}} &:\equiv \bot \\
(\varphi \wedge \psi)^{\mathcal{N}} &:\equiv \varphi^{\mathcal{N}} \wedge \psi^{\mathcal{N}} \\
(\varphi \Rightarrow \psi)^{\mathcal{N}} &:\equiv \varphi^{\mathcal{N}} \Rightarrow \psi^{\mathcal{N}} \\
(\varphi \vee \psi)^{\mathcal{N}} &:\equiv \neg\neg(\varphi^{\mathcal{N}} \vee \psi^{\mathcal{N}})
\end{aligned}
$$

The translation of the quantifiers is adapted from the case of predicate logic by recursively apply the translation also to the domain of quantification.

$$
\begin{aligned}
((\exists x \in A)\varphi)^{\mathcal{N}} &:\equiv \neg\neg(\exists x \in A^{\mathcal{N}})\varphi^{\mathcal{N}} \\
((\forall x \in A)\varphi)^{\mathcal{N}} &:\equiv (\forall x \in A^{\mathcal{N}})\varphi^{\mathcal{N}}
\end{aligned}
$$

Contrary to the case of predicate logic, we do not double-negate the equality predicates. We just translate them as follows.

$$
\mathsf{Eq}(A, a, b)^{\mathcal{N}} :\equiv \mathsf{Eq}(A^{\mathcal{N}}, a^{\mathcal{N}}, b^{\mathcal{N}})
$$

The burden of proving that the resulting proposition is stable will be transferred in the validity theorem to the translation of types, which will be required to always produce types with stable equality.

Finally, the true term is mapped to itself.

$$
\mathsf{true}^{\mathcal{N}} :\equiv \mathsf{true}
$$

Set constructors. Since we do not want to alter set-theoretic constructions, we just recursively apply the translation to their sub-expressions. We report here the cases of the empty set and dependent sums:

$$
\begin{aligned}
\mathsf{N_0}^{\mathcal{N}} &:\equiv \mathsf{N_0} \\
\mathsf{El}_{\mathsf{N_0}}(c)^{\mathcal{N}} &:\equiv \mathsf{El}_{\mathsf{N_0}}(c^{\mathcal{N}})
\end{aligned}
$$

$$
\begin{aligned}
((\Sigma x \in A)B)^{\mathcal{N}} &:\equiv (\Sigma x \in A^{\mathcal{N}})B^{\mathcal{N}} \\
\langle a, b\rangle^{\mathcal{N}} &:\equiv \langle a^{\mathcal{N}}, b^{\mathcal{N}}\rangle \\
\mathsf{El}_{\Sigma}(d, (x, y).m)^{\mathcal{N}} &:\equiv \mathsf{El}_{\Sigma}(d^{\mathcal{N}}, (x, y).m^{\mathcal{N}})
\end{aligned}
$$

The same (trivial) pattern will apply also to the pre-syntax of singleton set, disjoint sums, dependent products, lists and quotients.

Power-collection of the singleton. We translate the power collection of the singleton into its subcollection of stable propositions up to equiprovability.

$$\mathcal{P}(1)^{\mathcal{N}} :\equiv (\Sigma\, x \in \mathcal{P}(1))(\neg\neg \mathsf{Dc}(x) \Rightarrow \mathsf{Dc}(x))$$

The translation of its introduction constructor just accounts for this change.

$$[\varphi]^{\mathcal{N}} :\equiv \langle [\varphi^{\mathcal{N}}], \mathsf{true}\rangle$$

This concludes the definition of the translation. We immediately notice that it enjoys the following syntactical property, which we will tacitly exploit in the rest of the discussion.

Lemma 4.2.2 (Substitution)**.** *If e and t are two expressions of the pre-syntax, and x is a variable, then $(e[t/x])^{\mathcal{N}} \equiv e^{\mathcal{N}}[t^{\mathcal{N}}/x]$.*

Proof. Straightforward, by induction on the pre-syntax. □

Before proving the Validity Theorem, we recall a series of closure properties for stable propositions. The following statements are already true in intuitionistic predicate logic.

Proposition 4.2.3.

1. *$\bot$ is a stable proposition;*
2. *if φ and ψ are stable propositions, then also $\varphi \wedge \psi$ is;*
3. *if φ is a proposition (not necessarily stable) and ψ is a stable proposition, then also $\varphi \Rightarrow \psi$ is a stable proposition;*
4. *if $\varphi(x)$ prop $[x \in A]$ is a stable proposition, then so is $(\forall x \in A)\varphi(x)$;*
5. *a negated proposition $\neg\varphi$ is always stable.*

Proof. Entirely analogous to the case of intuitionistic predicate logic. □

The next proposition will be vital to prove point 2 of the Validity Theorem 4.2.5, as it shows how stability of the equality predicate is preserved by type constructors.

Proposition 4.2.4. *In* **emTT** *the following closure properties for type with stable equality hold:*

1. *the set N_0 has stable equality;*

2. *the set* N_1 *has stable equality;*

3. *if* A *is a set with stable equality, then* $\mathsf{List}(A)$ *has stable equality;*

4. *if* A *and* B *are sets with stable equality, then* $A + B$ *has stable equality;*

5. *if* A *is a type with stable equality and* $B(x)$ *is a dependent family over* A *with stable equality, then* $(\Sigma x \in A)B(x)$ *has stable equality;*

6. *if* A *is a set (not necessarily with stable equality) and* $B(x)$ *is a dependent family over* A *with stable equality, then if* $(\Pi x \in A)B(x)$ *is a well-formed collection, then it has stable equality;*

7. *if* A *is a set (not necessarily with stable equality) and* R *is a stable small equivalence relation over* A*, then* A/R *has stable equality;*

8. *the collection* $(\Sigma x \in \mathcal{P}(1))(\neg\neg\mathsf{Dc}(x) \Rightarrow \mathsf{Dc}(x))$ *has stable equality;*

9. *propositions have stable equality.*

Proof. In the following, in the proof of each point we implicitly exploit the characterisation of equality of type constructors proved in Proposition 1.7.8, and the usual properties of stable propositions recalled in Proposition 4.2.3.

1. Trivial, since $\bot$ is stable.

2. Trivial, since $\top$ is stable.

3. By induction on lists, again using the fact that $\bot$ and $\top$ are stable, and that conjunction preserves stability.

4. By induction on disjoint sums.

5. Assume A *col* and $B(x)$ *col* $[x \in A]$ to be two collections with stable equality. We must prove that for terms $a, a' \in A$, $b \in B(a)$, and $b' \in B(a')$ the proposition $(\exists x \in a =_A a')\ b =_{B(a)} b'$ is stable. By the elimination rule of the existential quantifier, we can derive the following.

$$(\exists x \in a =_A a')\ b =_{B(a)} b' \quad \Rightarrow \quad a =_A a'$$
$$(\forall y \in a =_A a')(\ (\exists x \in a =_A a')\ b =_{B(a)} b' \quad \Rightarrow \quad b =_{B(a)} b'\ )$$

From these, we get

$$\neg\neg(\exists x \in a =_A a')\, b =_{B(a)} b' \quad \Rightarrow \quad \neg\neg a =_A a' \tag{4.1}$$
$$(\forall y \in a =_A a')(\neg\neg(\exists x \in a =_A a')\, b =_{B(a)} b' \;\; \Rightarrow \;\; \neg\neg b =_{B(a)} b') \tag{4.2}$$

Assume $\neg\neg(\exists x \in a =_A a')\, b =_{B(a)} b'$; from (4.1) we deduce $\neg\neg a =_A a'$ and, since A has stable equality, we conclude $a =_A a'$; knowing that $a =_A a'$ holds, we can now apply the hypothesis to (4.2) and we deduce $\neg\neg b =_{B(a)} b'$, which, since $B(a)$ has stable equality for any given a, implies $b =_{B(a)} b'$; finally, by the introduction rule of the existential quantifier we have $(\exists x \in a =_A a')\, b =_{B(a)} b'$. Hence, we have shown that the proposition $(\exists x \in a =_A a')\, b =_{B(a)} b'$ is stable.

6. Stability is preserved by universal quantification.

7. By induction on quotients, recalling that quotients in **emTT** are effective.

8. For the collection $(\Sigma x \in \mathcal{P}(1))(\neg\neg \mathsf{Dc}(x) \Rightarrow \mathsf{Dc}(x))$ we have that, by the rules for equality of dependent pairs and propositions in Proposition 1.7.8, its propositional equality is equivalent to

$$\pi_1(z) =_{\mathcal{P}(1)} \pi_1(w) \quad \text{with } z, w \in (\Sigma x \in \mathcal{P}(1))(\neg\neg \mathsf{Dc}(x) \Rightarrow \mathsf{Dc}(x))$$

which, again by the case of $\mathcal{P}(1)$ in Proposition 1.7.8, is in turn equivalent to

$$\mathsf{Dc}(\pi_1(z)) \Leftrightarrow \mathsf{Dc}(\pi_1(w))$$

Since the propositions $\mathsf{Dc}(\pi_1(z))$ and $\mathsf{Dc}(\pi_1(w))$ are stable (by $\pi_2(z)$ and $\pi_2(w)$, respectively), and since conjunction and implication preserve stability, we conclude that $(\Sigma x \in \mathcal{P}(1))(\neg\neg \mathsf{Dc}(x) \Rightarrow \mathsf{Dc}(x))$ has stable equality.

9. Trivial, since $\top$ is stable.

□

We are now ready to prove the validity of the interpretation.

Theorem 4.2.5 (Validity)**.** *The translation is an interpretation of* **emTT**^{c} *into* **emTT***, in the sense that it preserves judgement derivability between the two theories:*

1. *if* $\mathbf{emTT}^c \vdash \Gamma\ ctx$, *then* $\mathbf{emTT} \vdash \Gamma^{\mathcal{N}}\ ctx$
2. *if* $\mathbf{emTT}^c \vdash A\ type\ [\Gamma]$, *then* $\mathbf{emTT} \vdash A^{\mathcal{N}}\ type\ [\Gamma^{\mathcal{N}}]$
3. *if* $\mathbf{emTT}^c \vdash a \in A\ [\Gamma]$, *then* $\mathbf{emTT} \vdash a^{\mathcal{N}} \in A^{\mathcal{N}}\ [\Gamma^{\mathcal{N}}]$
4. *if* $\mathbf{emTT}^c \vdash A = B\ type\ [\Gamma]$, *then* $\mathbf{emTT} \vdash A^{\mathcal{N}} = B^{\mathcal{N}}\ type\ [\Gamma^{\mathcal{N}}]$
5. *if* $\mathbf{emTT}^c \vdash a = b \in A\ [\Gamma]$, *then* $\mathbf{emTT} \vdash a^{\mathcal{N}} = b^{\mathcal{N}} \in A^{\mathcal{N}}\ [\Gamma^{\mathcal{N}}]$

Moreover, the translation produces stable propositions and, in particular, collections with stable equality:

6. *if* $\mathbf{emTT}^c \vdash \varphi\ prop\ [\Gamma]$, *then*

$$\mathbf{emTT} \vdash \neg\neg\varphi^{\mathcal{N}} \Rightarrow \varphi^{\mathcal{N}}\ \mathsf{true}\ [\Gamma^{\mathcal{N}}]$$

7. *if* $\mathbf{emTT}^c \vdash A\ col\ [\Gamma]$, *then*

$$\mathbf{emTT} \vdash \neg\neg\mathsf{Eq}(A^{\mathcal{N}}, x, y) \Rightarrow \mathsf{Eq}(A^{\mathcal{N}}, x, y)\ \mathsf{true}\ [\Gamma^{\mathcal{N}}, x \in A^{\mathcal{N}}, y \in A^{\mathcal{N}}]$$

Proof. All seven points are proved simultaneously by induction on judgements derivation. Due to the trivial pattern of the translation on most of the constructors, the majority of cases are trivially checked.

The cases involving the axiom of the excluded middle, the falsum constant, the disjunction and the existential quantifier are checked as in the case of translating classical predicate logic into the intuitionistic one using Proposition 4.2.3.

Point 6 on propositional equality is checked using the inductive hypothesis on point 7; in turn, for point 7 it suffices to apply the inductive hypotheses using Proposition 4.2.4. □

As an immediate corollary we have that

Corollary 4.2.6. *The theories* $\mathbf{emTT}^c$ *and* $\mathbf{emTT}$ *are equiconsistent.*

Proof. By point 3 of Theorem 3.4.3, since the inconsistency judgement $\mathsf{true} \in \bot\ [\,]$ is sent by the translation to itself. □

4.3 Compatibility with classical predicativism à la Weyl

As anticipated in Section 1.2.3, from the equiconsistency between **emTT** and **emTT**^{c}, we can deduce that, accordingly to Weyl's treatment of classical predicative mathematics [Wey18], neither Dedekind real numbers nor functional relations from the set of natural numbers to itself form a set.

We start by observing that, although classical, in **emTT**^{c} the type of booleans $\mathsf{Bool} :\equiv \mathsf{N}_1 + \mathsf{N}_1$ is not a propositional classifier; this is because, even in the presence of the excluded middle, we cannot eliminate from a disjunction $\varphi \vee \neg\varphi$ towards the set $\mathsf{N}_1 + \mathsf{N}_1$. More generally, we can easily adapt Proposition 1.7.4 to prove the following result.

Proposition 4.3.1. *In* **emTT**^{c}*, the collections* $\mathcal{P}(1)$ *and* $\mathcal{P}(\mathbb{N})$ *are proper.*

Proof. Since we know by Corollary 4.2.6 that **emTT**^{c} is equiconsistent with **emTT**, we can follow the same proof of Proposition 1.7.4. □

Moreover, recall that in **emTT** the collection of real numbers can be defined through Dedekind (left) cuts as

$$\begin{aligned}\mathbb{R} :\equiv (\Sigma A \in \mathcal{P}(\mathbb{Q}))((\exists q \in \mathbb{Q})q \,\varepsilon\, A \\ &\wedge (\exists q \in \mathbb{Q})\neg q \,\varepsilon\, A \\ &\wedge (\forall q \,\varepsilon\, A)(\forall r \in \mathbb{Q})(r < q \Rightarrow r \,\varepsilon\, A) \\ &\wedge (\forall q \,\varepsilon\, A)(\exists r \,\varepsilon\, A)q < r)\end{aligned}$$

while the collection of functional relations from the set of natural numbers to itself as

$$\mathsf{FunRel}(\mathbb{N},\mathbb{N}) :\equiv (\Sigma R \in \mathcal{P}(\mathbb{N}\times\mathbb{N}))(\forall x \in \mathbb{N})(\exists! y \in \mathbb{N})R(\langle x, y\rangle)$$

Theorem 4.3.2. *In* **emTT**^{c}*, and thus also in* **emTT***, the collections* $\mathbb{R}$ *and* $\mathsf{FunRel}(\mathbb{N},\mathbb{N})$ *are proper.*

Proof. If $\mathbb{R}$ were isomorphic to a set, then the set $\{0,1\}_{\mathbb{R}}$ obtained from $\mathbb{R}$ by comprehension through the small proposition

$$(\forall q \in \mathbb{Q})(q \,\varepsilon\, A \Leftrightarrow q < 0) \vee (\forall q \in \mathbb{Q})(q \,\varepsilon\, A \Leftrightarrow q < 1) \text{ with } A \in \mathcal{P}(\mathbb{Q})$$

would be isomorphic to a set too. In turn, it is easy to show that, classically, $\mathcal{P}(1)$ is isomorphic to $\{0,1\}_{\mathbb{R}}$; the isomorphism is obtained by specialising to $\{0,1\}_{\mathbb{R}}$ the following operations between $\mathcal{P}(1)$ and $\mathcal{P}(\mathbb{Q})$.

$$[(\forall q \in \mathbb{Q})(q \,\varepsilon\, A \Leftrightarrow q < 1)] \in \mathcal{P}(1) \qquad [A \in \mathcal{P}(\mathbb{Q})]$$
$$\{q \in \mathbb{Q} \,|\, (q < 0 \wedge \neg \mathsf{Dc}(x)) \vee (q < 1 \wedge \mathsf{Dc}(x))\} \in \mathcal{P}(\mathbb{Q}) \qquad [x \in \mathcal{P}(1)]$$

We conclude by Proposition 4.3.1.

For $\mathsf{FunRel}(\mathbb{N},\mathbb{N})$ the proof is analogous, using the set obtained by comprehension from it through the small proposition $R(\langle x,y\rangle) \Rightarrow y =_{\mathbb{N}} 0 \vee y =_{\mathbb{N}} 1$, with $R \in \mathcal{P}(\mathbb{N}\times\mathbb{N})$. □

4.4 Equiconsistency of the Calculus of Constructions with its classical version

Recall that the intensional level **mTT** of **MF** can be seen as a predicative version of the Calculus of Constructions in [CH88]. More precisely, consider the impredicative theory $\mathbf{mTT}_{\mathsf{imp}}$ obtained by extending the intensional level **mTT** with the congruence rules for types and terms and with the following resizing rules collapsing the predicative distinction between effective and open-ended types.

$$\textsf{col-into-set}\ \frac{A\ col}{A\ set} \qquad \textsf{prop-into-prop}_{\mathsf{s}}\ \frac{\varphi\ prop}{\varphi\ prop_s}$$

Analogously, consider the impredicative theory $\mathbf{emTT}_{\mathsf{imp}}$ obtained by extending **emTT** with the same resizing rules above.

The theories $\mathbf{mTT}_{\mathsf{imp}}$ and $\mathbf{emTT}_{\mathsf{imp}}$ can be interpreted as an extended version of the Calculus of Constructions with inductive types from **MLTT**, and an extensional version of it with the quotient constructor, respectively. In particular, we know from Theorem 1.8.2, that $\mathbf{mTT}_{\mathsf{imp}}$ coincides with $\mathbf{CC}_{\mathsf{ML}}$.

Remarkably, the addition of impredicativity to **MF** does not affect most of the techniques used to investigate its meta-mathematical properties. In particular, the quotient interpretation of [Mai09], the equiconsistency of the two levels proven in Chapter 3, and the equiconsistency with the classical version proven in this chapter all scale easily to the impredicative case.

Proposition 4.4.1. *The theory* $\mathbf{emTT}_{\mathsf{imp}}$ *is interpretable in the quotient model constructed over* $\mathbf{mTT}_{\mathsf{imp}}$.

Proof. By using the same interpretation defined in [Mai09]. The additional resizing rules of $\mathbf{emTT}_{\mathsf{imp}}$ are easily validated. For example, consider the rule $\mathsf{col\text{-}into\text{-}set}$; to check its validity we need to know that, for each $\mathbf{emTT}_{\mathsf{imp}}$-collection A, the dependent extensional collection $A^{\mathcal{I}}_{=}$ interpreting it is a dependent extensional set, which, by definition, amounts to know that its support $A^{\mathcal{I}}$ is a set and its equivalence relation $=_{A^{\mathcal{I}}}$ is a small proposition; but this is guaranteed precisely by the resizing rules of $\mathbf{mTT}_{\mathsf{imp}}$. □

Corollary 4.4.2. *The theory* $\mathbf{emTT}_{\mathsf{imp}}$ *is interpretable in the quotient model constructed over* $\mathbf{CC}_{\mathsf{ML}}$.

Proof. Combining Theorem 1.8.2 and 4.4.1. □

Proposition 4.4.3. $\mathbf{emTT}_{\mathsf{imp}} + \mathsf{propext}$ *is conservative over* $\mathbf{emTT}_{\mathsf{imp}}$*, and* $\mathbf{emTT}^{c}_{\mathsf{imp}} + \mathsf{propext}$ *is conservative over* $\mathbf{emTT}^{c}_{\mathsf{imp}}$.

Proof. Since canonical isomorphisms has been defined inductively in the meta-theory, and not internally as in [CM22], we can use the same interpretation described in Definition 3.4.1. In the second point of the Validity Theorem 3.4.3, the additional resizing rules of the source theories are validated thanks to the same rules in the corresponding target theory; in the third point of the same theorem, the additional axiom LEM is validated analogously, thanks to the fact that the interpretation fixes the connectives: $[\![\varphi \vee \neg\varphi]\!] \equiv \ulcorner\varphi \vee \neg\varphi\urcorner$ whenever $[\![\varphi]\!] \equiv \ulcorner\varphi\urcorner$. By the same observations, Proposition 3.4.6 still holds in the presence of resizing rules and of LEM. Then, we can conclude as in Corollary 3.4.7. □

Proposition 4.4.4. *The theories* $\mathbf{emTT}^{c}_{\mathsf{imp}}$ *and* $\mathbf{emTT}_{\mathsf{imp}}$ *are equiconsistent.*

Proof. By using the double-negation interpretation already defined in 4.2.1 for the predicative case; the additional resizing rules are trivially validated in the second point of the Validity Theorem 4.2.5. □

We then consider the *classical version of* $\mathbf{CC}_{\mathsf{ML}}$ obtained by adding to its calculus a constant lem formalising the Law of the Excluded Middle.

$$\mathsf{lem} \in (\forall x \in \mathsf{Prop})(x \vee \neg x)$$

We call $\mathbf{CC}^c_{\mathsf{ML}}$ the resulting theory. Notice that, contrary to **MF**, where we focused on the extensional level to define its classical version, here we chose to add classical logic directly in the intensional level. We think this choice is more in line with the existing literature on classical extensions of the Calculus of (Inductive) Constructions.

Remark 4.4.5. Even with the additional power given by the combination of impredicativity and the excluded middle, in $\mathbf{CC}^c_{\mathsf{ML}}$, the set Bool is not a propositional classifier, again because of the distinction between propositions and arbitrary types.

Proposition 4.4.6. *$\mathbf{CC}^c_{\mathsf{ML}}$ is interpretable in $\mathbf{emTT}^c_{\mathsf{imp}} + \mathsf{propext}$.*

Proof. Thanks to Theorem 1.8.2, we can refer to the theory $\mathbf{mTT}_{\mathsf{imp}}$ extended with the constant lem above. Then, we extend the interpretation of Proposition 3.2.1 by sending such new constant to the canonical proof-term of the extensional level $\mathsf{lem} \mapsto \mathsf{true}$. The additional rules assumed on top of those of **mTT**, namely the congruence rules, the resizing rules, and the typing axiom of lem are validated by the interpretation simply because all their translations are equivalent to rules already present in $\mathbf{emTT}^c_{\mathsf{imp}}$. □

Corollary 4.4.7. *The theories $\mathbf{CC}_{\mathsf{ML}}$ and $\mathbf{CC}^c_{\mathsf{ML}}$ are equiconsistent.*

Proof. Following the chain of interpretations depicted below, successively applying Proposition 4.4.6, Proposition 4.4.3, Proposition 4.4.4, and Corollary 4.4.2.

$$\begin{array}{ccccc} \mathbf{CC}^c_{\mathsf{ML}} & \dashrightarrow & & & \mathbf{CC}_{\mathsf{ML}} \\ \downarrow & & & & \uparrow \\ \mathbf{emTT}^c_{\mathsf{imp}} + \mathsf{propext} & \longrightarrow & \mathbf{emTT}^c_{\mathsf{imp}} & \longrightarrow & \mathbf{emTT}_{\mathsf{imp}} \end{array}$$

□

Conclusions

In this work, we first provided a thorough introduction to the Minimalist Foundation, recalling in particular its compatibility with other foundational systems for constructive mathematics (Chapter 1). Drawing from the works in [MS23a] and [Sab24], these compatibility results were then extended to the presence of inductive and coinductive predicates, which in turn we have shown to coincide with the inductive and coinductive construction of topologies in Formal Topology and the well-known (co)inductive scheme of (non-)well-founded trees in Martin-Löf's type theory (Chapter 2). Successively, we proceeded to investigate some important meta-theoretical properties of the Minimalist Foundation by adapting the results in [MS24]. On the one hand, we improved our understanding of the relationship between its two levels by proving their equiconsistency, and, more generally, their equivalence over second-order arithmetic (Chapter 3). On the other hand, we investigated the classical version of the Minimalist Foundation, showing, in particular, its equiconsistency with the standard intuitionistic version and its compatibility with classical predicativism à la Weyl (Chapter 4).

As future works, we would especially like to merge the results of Chapters 2 and 4 in the sense of proving the equiconsistency between the classical and the intuitionistic version of the Minimalist Foundation extended with inductive and coinductive predicates. The difficulty of this task lies in the fact that we cannot reuse the same technique we used for the base calculus, namely the Gödel-Gentzen double-negation translation; since it does not go through the (co)inductive definitions considered in Chapter 2.

Bibliography

[AAG05] M. G. Abbott, T. Altenkirch, and N. Ghani. “Containers: Constructing strictly positive types”. In: *Theor. Comput. Sci.* 342.1 (2005), pp. 3–27.

[ACS15] B. Ahrens, P. Capriotti, and R. Spadotti. “Non-wellfounded trees in homotopy type theory”. In: *13th International Conference on Typed Lambda Calculi and Applications*. Vol. 38. LIPIcs. Leibniz Int. Proc. Inform. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2015, pp. 17–30. ISBN: 978-3-939897-87-3.

[Acz77] P. Aczel. “An Introduction to Inductive Definitions”. In: *Handbook of Mathematical Logic*. Ed. by Jon Barwise. Vol. 90. Studies in Logic and the Foundations of Mathematics. Elsevier, 1977, pp. 739–782. DOI: `https://doi.org/10.1016/S0049-237X(08)71120-0`. URL: `https://www.sciencedirect.com/science/article/pii/S0049237X08711200`.

[AGMM15] T. Altenkirch, N. Ghani, C. McBride, and P. Morris. “Indexed containers”. In: *Journal of Functional Programming* 25 (2015), e5. DOI: `10.1017/S095679681500009X`.

[AK16] Thorsten Altenkirch and Ambrus Kaposi. “Type Theory in Type Theory Using Quotient Inductive Types”. In: *SIGPLAN Not.* 51.1 (Jan. 2016), pp. 18–29. ISSN: 0362-1340. DOI: `10.1145/2914770.2837638`. URL: `https://doi.org/10.1145/2914770.2837638`.

[AR10] P. Aczel and M. Rathjen. “Notes on constructive set theory”. Book draft. 2010. URL: `http://www1.maths.leeds.ac.uk/~rathjen/book.pdf`.

[BB12] E. Bishop and D. Bridges. *Constructive Analysis*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2012. ISBN: 9783642616679. URL: https://books.google.it/books?id=GF8lBAAAQBAJ.

[Bre15] L. Bressan. *An extension of the Minimalist Foundation with well founded trees*. University of Padova, supervised by M. E. Maietti, 2015.

[CH88] T. Coquand and G. P. Huet. “The Calculus of Constructions”. In: *Inf. Comput.* 76 (1988), pp. 95–120. URL: https://api.semanticscholar.org/CorpusID:15433820.

[CM22] M. Contente and M. E. Maietti. “The Compatibility of the Minimalist Foundation with Homotopy Type Theory”. In: 2022. URL: https://api.semanticscholar.org/CorpusID:250408058.

[CM24] M. Contente and M. E. Maietti. *On the Compatibility of Constructive Predicative Mathematics with Weyl’s Classical Predicativity*. 2024. arXiv: 2407.04161 [math.LO]. URL: https://arxiv.org/abs/2407.04161.

[CP90] T. Coquand and C. Paulin. “Inductively defined types”. In: *COLOG-88*. Ed. by Per Martin-Löf and Grigori Mints. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, pp. 50–66. ISBN: 978-3-540-46963-6.

[CS19] F. Ciraulo and G. Sambin. “Reducibility, a constructive dual of spatiality”. In: *J. Log. Anal.* 11 (2019), Paper No. FT1, 26. ISSN: 1759-9008.

[CSSV03] T. Coquand, G. Sambin, J. Smith, and S. Valentini. “Inductively generated formal topologies.” In: *Annals of Pure and Applied Logic* 124.1-3 (2003), pp. 71–106.

[CT20] T. Coquand and A. Tosun. “Formal Topology and Univalent Foundations”. In: *Proof and Computation II*. 2020. Chap. Chapter 6, pp. 255–266. DOI: 10.1142/9789811236488_0006. eprint: https://www.worldscientific.com/doi/pdf/10.1142/9789811236488_0006. URL: https://www.worldscientific.com/doi/abs/10.1142/9789811236488_0006.

[Dyb96] P. Dybjer. "Internal type theory". In: *Types for Proofs and Programs*. Ed. by Stefano Berardi and Mario Coppo. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 120–134. ISBN: 978-3-540-70722-6.

[Emm21] Jacopo Emmenegger. "W-types in setoids". In: *Logical Methods in Computer Science* Volume 17, Issue 3 (Sept. 2021). DOI: `10.46298/lmcs-17(3:28)2021`. URL: `https://lmcs.episciences.org/8521`.

[Fef82] S. Feferman. "Iterated inductive fixed-point theories: application to Hancock's conjecture". In: *Patras Logic Symposion*. Ed. by G. Metakides. North Holland, 1982, pp. 171–196.

[GA06] N. Gambino and P. Aczel. "The generalised type-theoretic interpretation of constructive set theory". In: *Journal of Symbolic Logic* 71.1 (2006), pp. 67–103. DOI: `10.2178/jsl/1140641163`.

[GH04] N. Gambino and M. Hyland. "Wellfounded trees and dependent polynomial functors". In: *Types for proofs and programs*. Vol. 3085. Lecture Notes in Comput. Sci. Springer, Berlin, 2004, pp. 210–225. ISBN: 3-540-22164-6. DOI: `10.1007/978-3-540-24849-1\_14`. URL: `https://doi.org/10.1007/978-3-540-24849-1_14`.

[Gim96] E. Giménez. "A Calculus of Infinite Constructions and its application to the verification of communicating systems". PhD thesis. Ecole Normale Supérieure de Lyon, 1996.

[Hof95] M. Hofmann. "Conservativity of equality reflection over intensional type theory". In: *International Workshop on Types for Proofs and Programs*. Springer. 1995, pp. 153–164.

[Hug17] J. Hugunin. *IWTypes*. `https://github.com/jashug/IWTypes`. 2017.

[Hug21] J. Hugunin. "Why Not W?" In: *26th International Conference on Types for Proofs and Programs (TYPES 2020)*. Ed. by Ugo de'Liguoro, Stefano Berardi, and Thorsten Altenkirch. Vol. 188. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, 8:1–8:9. ISBN: 978-3-95977-182-5. DOI: `10.4230/LIPIcs.`

TYPES.2020.8. URL: https://drops.dagstuhl.de/opus/volltexte/2021/13887.

[IMMS18] H. Ishihara, M. E. Maietti, S Maschio, and T. Streicher. "Consistency of the intensional level of the Minimalist Foundation with Church's thesis and axiom of choice". In: *Arch. Math. Logic* 57.7-8 (2018), pp. 873–888. ISSN: 0933-5846. DOI: 10.1007/s00153-018-0612-9. URL: https://doi.org/10.1007/s00153-018-0612-9.

[Kle52] S. C. Kleene. *Introduction to Metamathematics*. Princeton, New Jersey: D. van Nostrand, 1952.

[Law69] F. W. Lawvere. "Adjointness in Foundations". In: *Dialectica* 23.3/4 (1969), pp. 281–296. ISSN: 00122017, 17468361. URL: http://www.jstor.org/stable/42969800 (visited on 07/09/2024).

[Mai05] M. E. Maietti. "Modular correspondence between dependent type theories and categories including pretopoi and topoi". In: *Math. Structures Comput. Sci.* 15.6 (2005), pp. 1089–1149. ISSN: 0960-1295,1469-8072. DOI: 10.1017/S0960129505004962. URL: https://doi.org/10.1017/S0960129505004962.

[Mai09] M. E. Maietti. "A minimalist two-level foundation for constructive mathematics". In: *Ann. Pure Appl. Logic* 160.3 (2009), pp. 319–354. ISSN: 0168-0072,1873-2461. DOI: 10.1016/j.apal.2009.01.006. URL: https://doi.org/10.1016/j.apal.2009.01.006. See the revised version at URL: https://arxiv.org/abs/0811.2774.

[Mai17] M. E. Maietti. "On choice rules in dependent type theory". In: *Theory and applications of models of computation*. Vol. 10185. Lecture Notes in Comput. Sci. Springer, Cham, 2017, pp. 12–23. DOI: 10.1007/978-3-319-55911-7_2. URL: https://doi.org/10.1007/978-3-319-55911-7_2.

[Mai19] M. E. Maietti. "Discriminating between foundations for neutral constructivism via the Minimalist Foundation". Mathematical Logic and Constructivity: The Scope and Limits of Neutral Constructivism. 2019. URL: http://logic.math.su.se/mloc-2019/programme.html#maria-emilia-maietti-abstract.

[Mai20] M. E. Maietti. “What notion of predicative topos?” Memorial Conference for Erik Palmgren. 2020. URL: http://logic.math.su.se/palmgren-memorial/#maietti.

[Mar84] P. Martin-Löf. *Intuitionistic Type Theory, notes by G. Sambin of a series of lectures given in Padua, June 1980*. Bibliopolis, Naples, 1984.

[MM15] M. E. Maietti and S. Maschio. “An extensional Kleene realizability semantics for the Minimalist Foundation”. In: *20th International Conference on Types for Proofs and Programs*. Vol. 39. LIPIcs. Leibniz Int. Proc. Inform. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2015, pp. 162–186.

[MMR21] M. E. Maietti, S. Maschio, and M. Rathjen. “A realizability semantics for inductive formal topologies, Church’s thesis and axiom of choice”. In: *Log. Methods Comput. Sci.* 17.2 (2021), Paper No. 21, 21. ISSN: 1860-5974. URL: https://doi.org/10.23638/LMCS-17(2:21)2021.

[MMR22] M. E. Maietti, S. Maschio, and M. Rathjen. “Inductive and coinductive topological generation with Church’s thesis and the axiom of choice”. In: *Log. Methods Comput. Sci.* 18.4 (2022), Paper No. 5, 28. ISSN: 1860-5974.

[MP00] I. Moerdijk and E. Palmgren. “Wellfounded trees in categories”. In: *Annals of Pure and Applied Logic* 104.1 (2000), pp. 189–218. ISSN: 0168-0072. DOI: https://doi.org/10.1016/S0168-0072(00)00012-9. URL: https://www.sciencedirect.com/science/article/pii/S0168007200000129.

[MPR23] M. E. Maietti, F. Pasquali, and G. Rosolini. *Quasitoposes as elementary quotient completions*. 2023. arXiv: 2111.15299 [math.LO]. URL: https://arxiv.org/abs/2111.15299.

[MS05] M. E. Maietti and G. Sambin. “Toward a minimalist foundation for constructive mathematics”. In: *From sets and types to topology and analysis*. Vol. 48. Oxford Logic Guides. Oxford Univ. Press, Oxford, 2005, pp. 91–114. DOI: 10.1093/acprof:oso/9780198566519.003.0006. URL: https://doi.org/10.1093/acprof:oso/9780198566519.003.0006.

[MS22] S. Maschio and P. Sabelli. "On the compatibility between the minimalist foundation and constructive set theory". In: *Revolutions and revelations in computability*. Vol. 13359. Lecture Notes in Comput. Sci. Springer, Cham, 2022, pp. 172–185. DOI: 10.1007/978-3-031-08740-0_15. URL: https://doi.org/10.1007/978-3-031-08740-0_15.

[MS23a] M. E. Maietti and P. Sabelli. "A topological counterpart of well-founded trees in dependent type theory". In: *Electronic Notes in Theoretical Informatics and Computer Science* Volume 3 - Proceedings of MFPS XXXIX (Nov. 2023). DOI: 10.46298/entics.11755. URL: https://entics.episciences.org/11755.

[MS23b] M. E. Maietti and G. Sambin. "The minimalist foundation and Bishop's constructive mathematics". In: *Handbook of constructive mathematics*. Vol. 185. Encyclopedia Math. Appl. Cambridge Univ. Press, Cambridge, 2023, pp. 525–563. ISBN: 978-1-316-51086-5.

[MS24] M. E. Maietti and P. Sabelli. "Equiconsistency of the Minimalist Foundation with its classical version". In: *Annals of Pure and Applied Logic* (2024), p. 103524. ISSN: 0168-0072. DOI: https://doi.org/10.1016/j.apal.2024.103524. URL: https://www.sciencedirect.com/science/article/pii/S0168007224001283.

[MV04] M. E. Maietti and S. Valentini. "A Structural Investigation on Formal Topology: Coreflection of Formal Covers and Exponentiability". In: *The Journal of Symbolic Logic* 69.4 (2004), pp. 967–1005. ISSN: 00224812. URL: http://www.jstor.org/stable/30041774 (visited on 03/21/2024).

[NPS90] B. Nordström, K. Petersson, and J. Smith. *Programming in Martin Löf's Type Theory.* Clarendon Press, Oxford, 1990.

[Our05] N. Oury. "Extensionality in the Calculus of Constructions". In: *Theorem Proving in Higher Order Logics*. Ed. by Joe Hurd and Tom Melham. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 278–293. ISBN: 978-3-540-31820-0.

[Pal22] E. Palmgren. "From type theory to setoids and back". In: *Mathematical Structures in Computer Science* 32.10 (2022), pp. 1283–1312. DOI: 10.1017/S0960129521000189.

[PATW02] Pultr, Aleš, Tholen, and Walter. "Free Quillen factorization systems." eng. In: *Georgian Mathematical Journal* 9.4 (2002), pp. 807–820. URL: http://eudml.org/doc/50826.

[Péd24] P. M. Pédrot. ""Upon This Quote I Will Build My Church Thesis"". In: *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science*. LICS '24. Tallinn, Estonia: Association for Computing Machinery, 2024. DOI: 10.1145/3661814.3662070. URL: https://doi.org/10.1145/3661814.3662070.

[PS89] K. Petersson and D. Synek. "A set constructor for inductive sets in Martin-Löf's type theory". In: *Category theory and computer science (Manchester, 1989)*. Vol. 389. Lecture Notes in Comput. Sci. Springer, Berlin, 1989, pp. 128–140. DOI: 10.1007/BFb0018349. URL: https://doi.org/10.1007/BFb0018349.

[Rat05] M. Rathjen. "Generalized inductive definitions in constructive set theory". In: *From sets and types to topology and analysis*. Vol. 48. Oxford Logic Guides. Oxford Univ. Press, Oxford, 2005, pp. 23–40. DOI: 10.1093/acprof:oso/9780198566519.003.0002. URL: https://doi.org/10.1093/acprof:oso/9780198566519.003.0002.

[SA21] J. Sterling and C. Angiuli. "Normalization for Cubical Type Theory". In: *2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. 2021, pp. 1–15. DOI: 10.1109/LICS52264.2021.9470719.

[Sab24] P. Sabelli. *A topological reading of inductive and coinductive definitions in Dependent Type Theory*. 2024. arXiv: 2404.03494 [math.LO].

[Sam03] G. Sambin. "Some points in formal topology". In: *Theoretical Computer Science* 305 (2003), pp. 347–408.

[Sam19] G. Sambin. “Dynamics in Foundations: What Does It Mean in the Practice of Mathematics?” In: *Reflections on the Foundations of Mathematics: Univalent Foundations, Set Theory and General Thoughts*. Ed. by Stefania Centrone, Deborah Kant, and Deniz Sarikaya. Cham: Springer International Publishing, 2019, pp. 455–494. ISBN: 978-3-030-15655-8. DOI: `10.1007/978-3-030-15655-8_21`. URL: `https://doi.org/10.1007/978-3-030-15655-8_21`.

[Sam24] G. Sambin. *Positive Topology. A New Practice in Constructive Mathematic*. Oxford Logic Guides. Cary, NC: Oxford University Press, 2024.

[Sam87] G. Sambin. “Intuitionistic formal spaces - a first communication”. In: *Mathematical logic and its applications*. Plenum (1987), pp. 187–204.

[See84] R. A. G. Seely. “Locally cartesian closed categories and type theory”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 95.1 (1984), pp. 33–48. DOI: `10.1017/S0305004100061284`.

[Sel97] J. P. Seldin. “On the proof theory of Coquand’s calculus of constructions”. In: *Annals of Pure and Applied Logic* 83.1 (1997), pp. 23–101. ISSN: 0168-0072. DOI: `https://doi.org/10.1016/S0168-0072(96)00008-5`. URL: `https://www.sciencedirect.com/science/article/pii/S0168007296000085`.

[Sim09] Stephen G. Simpson. *Subsystems of Second Order Arithmetic*. 2nd ed. Perspectives in Logic. Cambridge University Press, 2009.

[Smi88] J. M. Smith. “The Independence of Peano’s Fourth Axiom from Martin-Löf’s Type Theory Without Universes”. In: *The Journal of Symbolic Logic* 53.3 (1988), pp. 840–845. ISSN: 00224812. URL: `http://www.jstor.org/stable/2274575` (visited on 06/25/2024).

[Spa23] M. Spadetto. *A conservativity result for homotopy elementary types in dependent type theory*. 2023. arXiv: `2303.05623 [math.LO]`.

[Str93] T. Streicher. “Investigations into intensional type theory”. PhD thesis. Habilitationsschrift, Ludwig-Maximilians-Universität München, 1993.

[SV98] G. Sambin and S. Valentini. “Building up a toolbox for Martin-Löf’s type theory: subset theory”. In: *Twenty-five years of constructive type theory (Venice, 1995)*. Vol. 36. Oxford Logic Guides. Oxford Univ. Press, New York, 1998, pp. 221–244. ISBN: 0-19-850127-7.

[TD88] A.S. Troelstra and D. van Dalen. *Constructivism in Mathematics, Vol 1*. ISSN. Elsevier Science, 1988. ISBN: 9780080570884. URL: `https://books.google.de/books?id=-tc2qp0-2bsC`.

[Uni13] The Univalent Foundations Program. *Homotopy type theory—univalent foundations of mathematics*. The Univalent Foundations Program, Princeton, NJ; Institute for Advanced Study (IAS), Princeton, NJ, 2013, pp. xiv+589.

[Wey18] H. Weyl. *Das Kontinuum. Kritische Untersuchungen über die Grundlagen der Analysis*. Berlin, Boston: De Gruyter, 1918. ISBN: 9783112451144. DOI: `doi:10.1515/9783112451144`. URL: `https://doi.org/10.1515/9783112451144`.

[WST19] T. Winterhalter, M. Sozeau, and N. Tabareau. “Eliminating reflection from type theory”. In: *Proceedings of the 8th ACM SIGPLAN International Conference on Certified Programs and Proofs* (2019). URL: `https://api.semanticscholar.org/CorpusID:57379755`.

Appendix A

Rules of the Calculi

A.1 Minimal Type Theory mTT

A.1.1 Structural rules

Inclusion between kinds

$$\mathsf{prop_s\text{-}into\text{-}prop}\ \frac{\varphi\ prop_s}{\varphi\ prop} \qquad \mathsf{prop_s\text{-}into\text{-}set}\ \frac{\varphi\ prop_s}{\varphi\ set}$$

$$\mathsf{set\text{-}into\text{-}col}\ \frac{A\ set}{A\ col} \qquad \mathsf{prop\text{-}into\text{-}col}\ \frac{\varphi\ prop}{\varphi\ col}$$

Context formation and variable assumption

$$\mathsf{ax}\ \frac{}{()\ ctx} \qquad \mathsf{F\text{-}ctx}\ \frac{A\ col\ [\Gamma]}{\Gamma, x \in A\ ctx}\ \text{(with } x \text{ fresh variable)}$$

$$\mathsf{var}\ \frac{\Gamma, x \in A, \Delta\ ctx}{x \in A\ [\Gamma, x \in A, \Delta]}$$

Equality rules

$$\mathsf{conv}\ \frac{a \in A \quad A = B\ type}{a \in B} \qquad \mathsf{eq\text{-}conv}\ \frac{a = b \in A \quad A = B\ type}{a = b \in B}$$

$$\mathsf{type\text{-}ref}\ \frac{A\ type}{A = A\ type} \qquad \mathsf{type\text{-}sym}\ \frac{B = A\ type}{A = B\ type}$$

$$\mathsf{type\text{-}tra}\ \frac{A = B\ type \quad B = C\ type}{A = C\ type}$$

$$\mathsf{term\text{-}ref}\ \frac{a \in A}{a = a \in A} \qquad \mathsf{term\text{-}sym}\ \frac{a = b \in A}{b = a \in A}$$

$$\mathsf{term\text{-}tra}\ \frac{a = b \in A \quad b = c \in A}{a = c \in A}$$

$$\mathsf{type\text{-}sub}\ \frac{C\ type\ [\Gamma, x \in A, \Delta] \quad a = b \in A\ [\Gamma]}{C[a/x] = C[b/x]\ type\ [\Gamma, \Delta[a/x]]}$$

$$\mathsf{term\text{-}sub}\ \frac{c \in C\ [\Gamma, x \in A, \Delta] \quad a = b \in A\ [\Gamma]}{c[a/x] = c[b/x] \in C[a/x]\ [\Gamma, \Delta[a/x]]}$$

A.1.2 Type constructors

Falsum

$$\text{F-}\bot\ \frac{}{\bot\ prop_s} \qquad \text{E-}\bot\ \frac{\varphi\ prop \quad p \in \bot}{\mathsf{El}_\bot(p) \in \varphi}$$

Leibniz Propositional Equality

$$\text{F-Id}\ \frac{A\ col \quad a \in A \quad b \in A}{\mathsf{Id}(A,a,b)\ prop} \qquad \text{F}_{\mathsf{s}}\text{-Id}\ \frac{A\ set \quad a \in A \quad b \in A}{\mathsf{Id}(A,a,b)\ prop_s}$$

$$\text{I-Id}\ \frac{a \in A}{\mathsf{id}(a) \in \mathsf{Id}(A,a,a)}$$

$$\text{E-Id}\ \frac{p \in \mathsf{Id}(A,a,b) \quad \varphi(x,y)\ prop\ [x \in A, y \in A] \quad t(x) \in \varphi(x,x)\ [x \in A]}{\mathsf{El}_{\mathsf{Id}}(p,(x).t) \in \varphi(a,b)}$$

$$\beta\text{C-Id}\ \frac{a \in A \quad \varphi(x,y)\ prop\ [x \in A, y \in A] \quad t(x) \in \varphi(x,x)\ [x \in A]}{\mathsf{El}_{\mathsf{Id}}(\mathsf{id}(a),(x).t) = t(a) \in \varphi(a,a)}$$

Conjunction

$$\text{F-}\wedge\ \frac{\varphi\ prop \quad \psi\ prop}{\varphi \wedge \psi\ prop} \qquad \text{F}_{\mathsf{s}}\text{-}\wedge\ \frac{\varphi\ prop_s \quad \psi\ prop_s}{\varphi \wedge \psi\ prop_s}$$

$$\text{I-}\wedge\ \frac{p \in \varphi \quad q \in \psi \quad \varphi\ prop \quad \psi\ prop}{\langle p,_\wedge q\rangle \in \varphi \wedge \psi}$$

$$\text{E}_1\text{-}\wedge\ \frac{p \in \varphi \wedge \psi}{\pi_1^\wedge(p) \in \varphi} \qquad \text{E}_2\text{-}\wedge\ \frac{p \in \varphi \wedge \psi}{\pi_2^\wedge(p) \in \psi}$$

$$\beta\text{C}_1\text{-}\wedge\ \frac{p \in \varphi \quad q \in \psi \quad \varphi\ prop \quad \psi\ prop}{\pi_1^\wedge(\langle p,_\wedge q\rangle) = p \in \varphi}$$

$$\beta\text{C}_2\text{-}\wedge\ \frac{p \in \varphi \quad q \in \psi \quad \varphi\ prop \quad \psi\ prop}{\pi_2^\wedge(\langle p,_\wedge q\rangle) = q \in \psi}$$

Disjunction

$$\mathsf{F}\text{-}\vee\ \frac{\varphi\ prop \quad \psi\ prop}{\varphi \vee \psi\ prop} \qquad \mathsf{F_s}\text{-}\vee\ \frac{\varphi\ prop_s \quad \psi\ prop_s}{\varphi \vee \psi\ prop_s}$$

$$\mathsf{I_1}\text{-}\vee\ \frac{p \in \varphi \quad \varphi\ prop \quad \psi\ prop}{\mathsf{inl}_\vee(p) \in \varphi \vee \psi} \qquad \mathsf{I_2}\text{-}\vee\ \frac{q \in \psi \quad \varphi\ prop \quad \psi\ prop}{\mathsf{inr}_\vee(q) \in \varphi \vee \psi}$$

$$\mathsf{E}\text{-}\vee\ \frac{p \in \varphi \vee \psi \quad \chi\ prop \quad t(x) \in \chi\ [x \in \varphi] \quad s(x) \in \chi\ [x \in \psi]}{\mathsf{El}_\vee(p, (x).t, (x).s) \in \chi}$$

$$\beta\mathsf{C_1}\text{-}\vee\ \frac{\varphi\ prop \quad \psi\ prop \quad \chi\ prop \quad p \in \varphi \quad t(x) \in \chi\ [x \in \varphi] \quad s(x) \in \chi\ [x \in \psi]}{\mathsf{El}_\vee(\mathsf{inl}_\vee(p), (x).t, (x).s) = t(p) \in \chi}$$

$$\beta\mathsf{C_2}\text{-}\vee\ \frac{\varphi\ prop \quad \psi\ prop \quad \chi\ prop \quad q \in \psi \quad t(x) \in \chi\ [x \in \varphi] \quad s(x) \in \chi\ [x \in \psi]}{\mathsf{El}_\vee(\mathsf{inr}_\vee(q), (x).t, (x).s) = s(q) \in \chi}$$

Implication

$$\mathsf{F}\text{-}\Rightarrow\ \frac{\varphi\ prop \quad \psi\ prop}{\varphi \Rightarrow \psi\ prop} \qquad \mathsf{F_s}\text{-}\Rightarrow\ \frac{\varphi\ prop_s \quad \psi\ prop_s}{\varphi \Rightarrow \psi\ prop_s}$$

$$\mathsf{I}\text{-}\Rightarrow\ \frac{\varphi\ prop \quad \psi\ prop \quad t(x) \in \psi\ [x \in \varphi]}{(\lambda_\Rightarrow x \in \varphi)t(x) \in \varphi \Rightarrow \psi}$$

$$\mathsf{E}\text{-}\Rightarrow\ \frac{p \in \varphi \Rightarrow \psi \quad q \in \varphi}{\mathsf{Ap}_\Rightarrow(p, q) \in \psi}$$

$$\beta\mathsf{C}\text{-}\Rightarrow\ \frac{\varphi\ prop \quad \psi\ prop \quad p \in \varphi \quad t(x) \in \psi\ [x \in \varphi]}{\mathsf{Ap}_\Rightarrow((\lambda_\Rightarrow x \in \varphi)t(x), p) = t(p) \in \psi}$$

Existential quantification

$$\mathsf{F}\text{-}\exists\ \frac{\varphi(x)\ prop\ [x \in A]}{(\exists x \in A)\varphi(x)\ prop} \qquad \mathsf{F_s}\text{-}\exists\ \frac{A\ set \quad \varphi(x)\ prop_s\ [x \in A]}{(\exists x \in A)\varphi(x)\ prop_s}$$

$$\mathsf{I}\text{-}\exists\ \frac{\varphi(x)\ prop\ [x \in A] \quad a \in A \quad p \in \varphi(a)}{\langle a,_\exists p\rangle \in (\exists x \in A)\varphi(x)}$$

$$\mathsf{E}\text{-}\exists\ \frac{p \in (\exists x \in A)\varphi(x) \quad \psi\ prop \quad t(x,y) \in \psi\ [x \in A, y \in \varphi(x)]}{\mathsf{El}_\exists(p,(x,y).t) \in \psi}$$

$$\beta\mathsf{C}\text{-}\exists\ \frac{\begin{array}{c}\varphi(x)\ prop\ [x \in A] \quad a \in A \quad p \in \varphi(a) \\ \psi\ prop \quad t(x,y) \in \psi\ [x \in A, y \in \varphi(x)]\end{array}}{\mathsf{El}_\exists(\langle a,_\exists p\rangle,(x,y).t) = t(a,p) \in \psi}$$

Universal quantification

$$\mathsf{F}\text{-}\forall\ \frac{\varphi(x)\ prop\ [x \in A]}{(\forall x \in A)\varphi(x)\ prop} \qquad \mathsf{F_s}\text{-}\forall\ \frac{A\ set \quad \varphi(x)\ prop_s\ [x \in A]}{(\forall x \in A)\varphi(x)\ prop_s}$$

$$\mathsf{I}\text{-}\forall\ \frac{\varphi(x)\ prop\ [x \in A] \quad t(x) \in \varphi(x)\ [x \in A]}{(\lambda_\forall x \in A)t(x) \in (\forall x \in A)\varphi(x)}$$

$$\mathsf{E}\text{-}\forall\ \frac{p \in (\forall x \in A)\varphi(x) \quad a \in A}{\mathsf{Ap}_\forall(p,a) \in \varphi(a)}$$

$$\beta\mathsf{C}\text{-}\forall\ \frac{\varphi(x)\ prop\ [x \in A] \quad t(x) \in \varphi(x)\ [x \in A] \quad a \in A}{\mathsf{Ap}_\forall((\lambda_\forall x \in A)t(x),a) = t(a) \in \varphi(a)}$$

Empty set

$$\mathsf{F}\text{-}\mathsf{N}_0\ \frac{}{\mathsf{N}_0\ set} \qquad \mathsf{E}\text{-}\mathsf{N}_0\ \frac{a \in \mathsf{N}_0 \quad B(x)\ col\ [x \in \mathsf{N}_0]}{\mathsf{El}_{\mathsf{N}_0}(a) \in B(a)}$$

Singleton set

$$\text{F-N}_1 \ \frac{}{\mathsf{N}_1 \ set} \qquad \text{I-N}_1 \ \frac{}{\star \in \mathsf{N}_1}$$

$$\text{E-N}_1 \ \frac{a \in \mathsf{N}_1 \quad B(x) \ col \ [x \in \mathsf{N}_1] \quad b \in B(\star)}{\mathsf{El}_{\mathsf{N}_1}(a, b) \in B(a)}$$

$$\beta\text{C-N}_1 \ \frac{B(x) \ col \ [x \in \mathsf{N}_1] \quad b \in B(\star)}{\mathsf{El}_{\mathsf{N}_1}(\star, b) = b \in B(\star)}$$

Universe of small propositions à la Tarski

$$\text{F-Prop}_\mathsf{s} \ \frac{}{\mathsf{Prop}_\mathsf{s} \ col} \qquad \text{I-Prop}_\mathsf{s} \ \frac{\varphi \ prop_s}{\widehat{\varphi} \in \mathsf{Prop}_\mathsf{s}} \qquad \text{E-Prop}_\mathsf{s} \ \frac{c \in \mathsf{Prop}_\mathsf{s}}{\mathsf{T}(c) \ prop_s}$$

$$\text{C-Prop}_\mathsf{s} \ \frac{\varphi \ prop_s}{\mathsf{T}(\widehat{\varphi}) = \varphi \ prop_s} \qquad \eta\text{-Prop}_\mathsf{s} \ \frac{c \in \mathsf{Prop}_\mathsf{s}}{\widehat{\mathsf{T}(c)} = c \in \mathsf{Prop}_\mathsf{s}}$$

$$\text{Eq-Prop}_\mathsf{s} \ \frac{\varphi = \psi \ prop_s}{\widehat{\varphi} = \widehat{\psi} \in \mathsf{Prop}_\mathsf{s}} \qquad \text{Eq-E-Prop}_\mathsf{s} \ \frac{c = d \in \mathsf{Prop}_\mathsf{s}}{\mathsf{T}(c) = \mathsf{T}(d) \ prop_s}$$

Dependent sums

$$\text{F-}\Sigma \ \frac{B(x) \ col \ [x \in A]}{(\Sigma x \in A)B(x) \ col} \qquad \text{F}_\mathsf{s}\text{-}\Sigma \ \frac{A \ set \quad B(x) \ set \ [x \in A]}{(\Sigma x \in A)B(x) \ set}$$

$$\text{I-}\Sigma \ \frac{a \in A \quad b \in B(a) \quad B(x) \ col \ [x \in A]}{\langle a, b \rangle \in (\Sigma x \in A)B(x)}$$

$$\text{E-}\Sigma \ \frac{\begin{array}{l} c \in (\Sigma x \in A)B(x) \\ M(z) \ col \ [z \in (\Sigma x \in A)B(x)] \quad m(x, y) \in M(\langle x, y \rangle) \ [x \in A, y \in B(x)] \end{array}}{\mathsf{El}_\Sigma(c, (x, y).m) \in M(c)}$$

$$\beta\text{C-}\Sigma \ \frac{\begin{array}{l} B(x) \ col \ [x \in A] \quad a \in A \quad b \in B(a) \\ M(z) \ col \ [z \in (\Sigma x \in A)B(x)] \quad m(x, y) \in M(\langle x, y \rangle) \ [x \in A, y \in B(x)] \end{array}}{\mathsf{El}_\Sigma(\langle a, b \rangle, (x, y).m) = m(a, b) \in M(\langle a, b \rangle)}$$

Dependent products

$$\mathsf{F}\text{-}\Pi\ \frac{A\ set \quad B(x)\ set\ [x \in A]}{(\Pi x \in A)B(x)\ set} \qquad \mathsf{F}\text{-}\Pi\mathsf{Prop_s}\ \frac{A\ set}{(\Pi x \in A)\mathsf{Prop_s}\ col}$$

$$\mathsf{I}\text{-}\Pi\ \frac{t(x) \in B(x)\ [x \in A] \quad (\Pi x \in A)B(x)\ col}{(\lambda x \in A)t(x) \in (\Pi x \in A)B(x)}$$

$$\mathsf{E}\text{-}\Pi\ \frac{f \in (\Pi x \in A)B(x) \quad a \in A}{\mathsf{Ap}(f, a) \in B(a)}$$

$$\beta\mathsf{C}\text{-}\Pi\ \frac{t(x) \in B(x)\ [x \in A] \quad (\Pi x \in A)B(x)\ col \quad a \in A}{\mathsf{Ap}((\lambda x \in A)t(x), a) = t(a) \in B(a)}$$

Disjoint sums

$$\mathsf{F}\text{-}+\ \frac{A\ set \quad B\ set}{A + B\ set}$$

$$\mathsf{I_1}\text{-}+\ \frac{a \in A \quad A + B\ set}{\mathsf{inl}(a) \in A + B} \qquad \mathsf{I_2}\text{-}+\ \frac{b \in B \quad A + B\ set}{\mathsf{inr}(b) \in A + B}$$

$$\mathsf{E}\text{-}+\ \frac{\begin{array}{l} c \in A + B \\ M(z)\ col\ [z \in A + B] \quad t(x) \in M(\mathsf{inl}(x))\ [x \in A] \quad s(x) \in M(\mathsf{inr}(x))\ [x \in B] \end{array}}{\mathsf{El}_+(c, (x).t, (x).s) \in M(c)}$$

$$\beta\mathsf{C_1}\text{-}+\ \frac{\begin{array}{l} a \in A \\ M(z)\ col\ [z \in A + B] \quad t(x) \in M(\mathsf{inl}(x))\ [x \in A] \quad s(x) \in M(\mathsf{inr}(x))\ [x \in B] \end{array}}{\mathsf{El}_+(\mathsf{inl}(a), (x).t, (x).s) = t(a) \in M(\mathsf{inl}(a))}$$

$$\beta\mathsf{C_2}\text{-}+\ \frac{\begin{array}{l} b \in B \\ M(z)\ col\ [z \in A + B] \quad t(x) \in M(\mathsf{inl}(x))\ [x \in A] \quad s(x) \in M(\mathsf{inr}(x))\ [x \in B] \end{array}}{\mathsf{El}_+(\mathsf{inr}(b), (x).t, (x).s) = s(b) \in M(\mathsf{inr}(b))}$$

Lists

$$\text{F-List}\ \frac{A\ set}{\mathsf{List}(A)\ set}$$

$$\text{I}_1\text{-List}\ \frac{A\ set}{\epsilon \in \mathsf{List}(A)} \qquad \text{I}_2\text{-List}\ \frac{l \in \mathsf{List}(A) \quad a \in A}{\mathsf{cons}(l,a) \in \mathsf{List}(A)}$$

$$\text{E-List}\ \frac{\begin{array}{l} l \in \mathsf{List}(A) \quad M(z)\ col\ [z \in \mathsf{List}(A)] \\ b \in M(\epsilon) \quad c(x,y,z) \in M(\mathsf{cons}(x,y))\ [x \in \mathsf{List}(A), y \in A, z \in M(x)] \end{array}}{\mathsf{El}_{\mathsf{List}}(l, b, (x,y,z).c) \in M(l)}$$

$$\beta\text{C}_1\text{-List}\ \frac{\begin{array}{l} M(z)\ col\ [z \in \mathsf{List}(A)] \\ b \in M(\epsilon) \quad c(x,y,z) \in M(\mathsf{cons}(x,y))\ [x \in \mathsf{List}(A), y \in A, z \in M(x)] \end{array}}{\mathsf{El}_{\mathsf{List}}(\epsilon, b, (x,y,z).c) = b \in M(\epsilon)}$$

$$\beta\text{C}_2\text{-List}\ \frac{\begin{array}{l} l \in \mathsf{List}(A) \quad a \in A \quad M(z)\ col\ [z \in \mathsf{List}(A)] \\ b \in M(\epsilon) \quad c(x,y,z) \in M(\mathsf{cons}(x,y))\ [x \in \mathsf{List}(A), y \in A, z \in M(x)] \end{array}}{\mathsf{El}_{\mathsf{List}}(\mathsf{cons}(l,a), b, (x,y,z).c) = c(l, a, \mathsf{El}_{\mathsf{List}}(l, b, (x,y,z).c)) \in M(\mathsf{cons}(l,a))}$$

A.2 Extensional Minimal Type Theory emTT

A.2.1 Structural rules

Inclusion between kinds

$$\text{prop}_\text{s}\text{-into-prop}\ \frac{\varphi\ prop_s}{\varphi\ prop} \qquad \text{prop}_\text{s}\text{-into-set}\ \frac{\varphi\ prop_s}{\varphi\ set}$$

$$\text{set-into-col}\ \frac{A\ set}{A\ col} \qquad \text{prop-into-col}\ \frac{\varphi\ prop}{\varphi\ col}$$

Context formation and variable assumption

$$\text{ax}\ \frac{}{()\ ctx} \qquad \text{F-ctx}\ \frac{A\ col\ [\Gamma]}{\Gamma, x \in A\ ctx}\ (\text{with } x \text{ fresh variable})$$

$$\text{var}\ \frac{\Gamma, x \in A, \Delta\ ctx}{x \in A\ [\Gamma, x \in A, \Delta]}$$

Equality rules

$$\text{eq-prop}_\text{s}\text{-into-prop}\ \frac{\varphi = \psi\ prop_s}{\varphi = \psi\ prop} \qquad \text{eq-prop}_\text{s}\text{-into-set}\ \frac{\varphi = \psi\ prop_s}{\varphi = \psi\ set}$$

$$\text{eq-set-into-col}\ \frac{A = B\ set}{A = B\ col} \qquad \text{eq-prop-into-col}\ \frac{\varphi = \psi\ prop}{\varphi = \psi\ col}$$

$$\text{conv}\ \frac{a \in A \quad A = B\ type}{a \in B} \qquad \text{eq-conv}\ \frac{a = b \in A \quad A = B\ type}{a = b \in B}$$

$$\text{type-ref}\ \frac{A\ type}{A = A\ type} \qquad \text{type-sym}\ \frac{B = A\ type}{A = B\ type}$$

$$\text{type-tra}\ \frac{A = B\ type \quad B = C\ type}{A = C\ type}$$

$$\text{term-ref}\ \frac{a \in A}{a = a \in A} \qquad \text{term-sym}\ \frac{a = b \in A}{b = a \in A}$$

$$\text{term-tra}\ \frac{a = b \in A \quad b = c \in A}{a = c \in A}$$

Proof irrelevance

$$\text{prop-true}\ \frac{\varphi\ prop \quad p \in \varphi}{\mathsf{true} \in \varphi} \qquad \text{prop-mono}\ \frac{\varphi\ prop \quad p \in \varphi \quad q \in \varphi}{p = q \in \varphi}$$

A.2.2 Type constructors

For readability, we postpone to the last paragraph the congruence rules of all type and term constructors. The only exception is for the congruence rules of the introduction term of the power-collection of the singleton and the introduction term of the quotients, which do not follow the common trivial pattern and are reported also in the corresponding section.

Falsum

$$\textsf{F-}\bot \ \frac{}{\bot \ prop_s} \qquad \textsf{E-}\bot \ \frac{\varphi \ prop \quad \bot \ true}{\varphi \ true}$$

Extensional Propositional Equality

$$\textsf{F-Eq} \ \frac{A \ col \quad a \in A \quad b \in A}{\textsf{Eq}(A, a, b) \ prop} \qquad \textsf{F}_\textsf{s}\textsf{-Eq} \ \frac{A \ set \quad a \in A \quad b \in A}{\textsf{Eq}(A, a, b) \ prop_s}$$

$$\textsf{I-Eq} \ \frac{a \in A}{\textsf{Eq}(A, a, a) \ true} \qquad \textsf{E-Eq} \ \frac{\textsf{Eq}(A, a, b) \ true}{a = b \in A}$$

Conjunction

$$\textsf{F-}\wedge \ \frac{\varphi \ prop \quad \psi \ prop}{\varphi \wedge \psi \ prop} \qquad \textsf{F}_\textsf{s}\textsf{-}\wedge \ \frac{\varphi \ prop_s \quad \psi \ prop_s}{\varphi \wedge \psi \ prop_s}$$

$$\textsf{I-}\wedge \ \frac{\varphi \ true \quad \psi \ true}{\varphi \wedge \psi \ true} \qquad \textsf{E}_1\textsf{-}\wedge \ \frac{\varphi \wedge \psi \ true}{\varphi \ true} \qquad \textsf{E}_2\textsf{-}\wedge \ \frac{\varphi \wedge \psi \ true}{\psi \ true}$$

Disjunction

$$\textsf{F-}\vee \ \frac{\varphi \ prop \quad \psi \ prop}{\varphi \vee \psi \ prop} \qquad \textsf{F}_\textsf{s}\textsf{-}\vee \ \frac{\varphi \ prop_s \quad \psi \ prop_s}{\varphi \vee \psi \ prop_s}$$

$$\textsf{I}_1\textsf{-}\vee \ \frac{\varphi \ true \quad \varphi \ prop \quad \psi \ prop}{\varphi \vee \psi \ true} \qquad \textsf{I}_2\textsf{-}\vee \ \frac{\psi \ true \quad \varphi \ prop \quad \psi \ prop}{\varphi \vee \psi \ true}$$

$$\textsf{E-}\vee \ \frac{\varphi \vee \psi \ true \quad \chi \ prop \quad \chi \ true \ [x \in \varphi] \quad \chi \ true \ [x \in \psi]}{\chi \ true}$$

Implication

$$\text{F-}\!\Rightarrow\ \frac{\varphi\ \mathit{prop}\quad \psi\ \mathit{prop}}{\varphi \Rightarrow \psi\ \mathit{prop}} \qquad \text{F}_\text{s}\text{-}\!\Rightarrow\ \frac{\varphi\ \mathit{prop}_s\quad \psi\ \mathit{prop}_s}{\varphi \Rightarrow \psi\ \mathit{prop}_s}$$

$$\text{I-}\!\Rightarrow\ \frac{\varphi\ \mathit{prop}\quad \psi\ \mathit{prop}\quad \psi\ \mathit{true}\ [x \in \varphi]}{\varphi \Rightarrow \psi\ \mathit{true}} \qquad \text{E-}\!\Rightarrow\ \frac{\varphi \Rightarrow \psi\ \mathit{true}\quad \varphi\ \mathit{true}}{\psi\ \mathit{true}}$$

Existential quantification

$$\text{F-}\exists\ \frac{\varphi(x)\ \mathit{prop}\ [x \in A]}{(\exists x \in A)\varphi(x)\ \mathit{prop}} \qquad \text{F}_\text{s}\text{-}\exists\ \frac{A\ \mathit{set}\quad \varphi(x)\ \mathit{prop}_s\ [x \in A]}{(\exists x \in A)\varphi(x)\ \mathit{prop}_s}$$

$$\text{I-}\exists\ \frac{\varphi(x)\ \mathit{prop}\ [x \in A]\quad a \in A\quad \varphi(a)\ \mathit{true}}{(\exists x \in A)\varphi(x)\ \mathit{true}}$$

$$\text{E-}\exists\ \frac{(\exists x \in A)\varphi(x)\ \mathit{true}\quad \psi\ \mathit{prop}\quad \psi\ \mathit{true}\ [x \in A, y \in \varphi(x)]}{\psi\ \mathit{true}}$$

Universal quantification

$$\text{F-}\forall\ \frac{\varphi(x)\ \mathit{prop}\ [x \in A]}{(\forall x \in A)\varphi(x)\ \mathit{prop}} \qquad \text{F}_\text{s}\text{-}\forall\ \frac{A\ \mathit{set}\quad \varphi(x)\ \mathit{prop}_s\ [x \in A]}{(\forall x \in A)\varphi(x)\ \mathit{prop}_s}$$

$$\text{I-}\forall\ \frac{\varphi(x)\ \mathit{prop}\ [x \in A]\quad \varphi(x)\ \mathit{true}\ [x \in A]}{(\forall x \in A)\varphi(x)\ \mathit{true}} \qquad \text{E-}\forall\ \frac{(\forall x \in A)\varphi(x)\ \mathit{true}\quad a \in A}{\varphi(a)\ \mathit{true}}$$

Empty set

$$\text{F-N}_0\ \frac{}{\mathsf{N}_0\ \mathit{set}} \qquad \text{E-N}_0\ \frac{a \in \mathsf{N}_0\quad B(x)\ \mathit{col}\ [x \in \mathsf{N}_0]}{\mathsf{El}_{\mathsf{N}_0}(a) \in B(a)}$$

Singleton set

$$\text{F-N}_1\ \frac{}{\mathsf{N}_1\ set} \qquad \text{I-N}_1\ \frac{}{\star \in \mathsf{N}_1}$$

$$\text{E-N}_1\ \frac{a \in \mathsf{N}_1 \quad B(x)\ col\ [x \in \mathsf{N}_1] \quad b \in B(\star)}{\mathsf{El}_{\mathsf{N}_1}(a, b) \in B(a)}$$

$$\beta\text{C-N}_1\ \frac{B(x)\ col\ [x \in \mathsf{N}_1] \quad b \in B(\star)}{\mathsf{El}_{\mathsf{N}_1}(\star, b) = b \in B(\star)}$$

Power-collection of the singleton

$$\text{F-}\mathcal{P}(1)\ \frac{}{\mathcal{P}(1)\ col} \qquad \text{I-}\mathcal{P}(1)\frac{\varphi\ prop_s}{[\varphi] \in \mathcal{P}(1)}$$

$$\text{eq-I-}\mathcal{P}(1)\frac{\varphi \Leftrightarrow \psi\ true}{[\varphi] = [\psi] \in \mathcal{P}(1)} \qquad \text{eff-}\mathcal{P}(1)\frac{[\varphi] = [\psi] \in \mathcal{P}(1)}{\varphi \Leftrightarrow \psi\ true}$$

$$\text{F}_\text{s}\text{-Eq-}\mathcal{P}(1)\ \frac{U \in \mathcal{P}(1) \quad V \in \mathcal{P}(1)}{\mathsf{Eq}(\mathcal{P}(1), U, V)\ prop_s} \qquad \eta\text{-}\mathcal{P}(1)\frac{U \in \mathcal{P}(1)}{[\mathsf{Eq}(\mathcal{P}(1), U, [\top])] = U \in \mathcal{P}(1)}$$

Dependent sums

$$\text{F-}\Sigma\ \frac{B(x)\ col\ [x \in A]}{(\Sigma x \in A)B(x)\ col} \qquad \text{F}_\text{s}\text{-}\Sigma\ \frac{A\ set \quad B(x)\ set\ [x \in A]}{(\Sigma x \in A)B(x)\ set}$$

$$\text{I-}\Sigma\ \frac{a \in A \quad b \in B(a) \quad B(x)\ col\ [x \in A]}{\langle a, b\rangle \in (\Sigma x \in A)B(x)}$$

$$\text{E-}\Sigma\ \frac{\begin{array}{l} c \in (\Sigma x \in A)B(x) \\ M(z)\ col\ [z \in (\Sigma x \in A)B(x)] \quad m(x,y) \in M(\langle x, y\rangle)\ [x \in A, y \in B(x)] \end{array}}{\mathsf{El}_\Sigma(c, (x,y).m) \in M(c)}$$

$$\beta\text{C-}\Sigma\ \frac{\begin{array}{l} B(x)\ col\ [x \in A] \quad a \in A \quad b \in B(a) \\ M(z)\ col\ [z \in (\Sigma x \in A)B(x)] \quad m(x,y) \in M(\langle x, y\rangle)\ [x \in A, y \in B(x)] \end{array}}{\mathsf{El}_\Sigma(\langle a, b\rangle, (x,y).m) = m(a,b) \in M(\langle a, b\rangle)}$$

Dependent products

$$\mathsf{F}\text{-}\Pi\ \frac{A\ set \quad B(x)\ set\ [x \in A]}{(\Pi x \in A)B(x)\ set} \qquad \mathsf{F}\text{-}\Pi\mathcal{P}(1)\ \frac{A\ set}{(\Pi x \in A)\mathcal{P}(1)\ col}$$

$$\mathsf{I}\text{-}\Pi\ \frac{t(x) \in B(x)\ [x \in A] \quad (\Pi x \in A)B(x)\ col}{(\lambda x \in A)t(x) \in (\Pi x \in A)B(x)}$$

$$\mathsf{E}\text{-}\Pi\ \frac{f \in (\Pi x \in A)B(x) \quad a \in A}{\mathsf{Ap}(f, a) \in B(a)}$$

$$\beta\mathsf{C}\text{-}\Pi\ \frac{t(x) \in B(x)\ [x \in A] \quad (\Pi x \in A)B(x)\ col \quad a \in A}{\mathsf{Ap}((\lambda x \in A)t(x), a) = t(a) \in B(a)}$$

$$\eta\mathsf{C}\text{-}\Pi\ \frac{f \in (\Pi x \in A)B(x)}{(\lambda x \in A)\mathsf{Ap}(f, x) = f \in (\Pi x \in A)B(x)}$$

Disjoint sums

$$\mathsf{F}\text{-}+\ \frac{A\ set \quad B\ set}{A + B\ set}$$

$$\mathsf{I}_1\text{-}+\ \frac{a \in A \quad A + B\ set}{\mathsf{inl}(a) \in A + B} \qquad \mathsf{I}_2\text{-}+\ \frac{b \in B \quad A + B\ set}{\mathsf{inr}(b) \in A + B}$$

$$\mathsf{E}\text{-}+\ \frac{\begin{array}{l} c \in A + B \\ M(z)\ col\ [z \in A + B] \quad t(x) \in M(\mathsf{inl}(x))\ [x \in A] \quad s(x) \in M(\mathsf{inr}(x))\ [x \in B] \end{array}}{\mathsf{El}_+(c, (x).t, (x).s) \in M(c)}$$

$$\beta\mathsf{C}_1\text{-}+\ \frac{\begin{array}{l} a \in A \\ M(z)\ col\ [z \in A + B] \quad t(x) \in M(\mathsf{inl}(x))\ [x \in A] \quad s(x) \in M(\mathsf{inr}(x))\ [x \in B] \end{array}}{\mathsf{El}_+(\mathsf{inl}(a), (x).t, (x).s) = t(a) \in M(\mathsf{inl}(a))}$$

$$\beta\mathsf{C}_2\text{-}+\ \frac{\begin{array}{l} b \in B \\ M(z)\ col\ [z \in A + B] \quad t(x) \in M(\mathsf{inl}(x))\ [x \in A] \quad s(x) \in M(\mathsf{inr}(x))\ [x \in B] \end{array}}{\mathsf{El}_+(\mathsf{inr}(b), (x).t, (x).s) = s(b) \in M(\mathsf{inr}(b))}$$

Lists

$$\text{F-List}\ \frac{A\ set}{\mathsf{List}(A)\ set}$$

$$\mathsf{I}_1\text{-List}\ \frac{A\ set}{\epsilon \in \mathsf{List}(A)} \qquad \mathsf{I}_2\text{-List}\ \frac{l \in \mathsf{List}(A) \quad a \in A}{\mathsf{cons}(l, a) \in \mathsf{List}(A)}$$

$$\text{E-List}\ \frac{\begin{array}{l} l \in \mathsf{List}(A) \quad M(z)\ col\ [z \in \mathsf{List}(A)] \\ b \in M(\epsilon) \quad c(x, y, z) \in M(\mathsf{cons}(x, y))\ [x \in \mathsf{List}(A), y \in A, z \in M(x)] \end{array}}{\mathsf{El}_{\mathsf{List}}(l, b, (x, y, z).c) \in M(l)}$$

$$\beta\mathsf{C}_1\text{-List}\ \frac{\begin{array}{l} M(z)\ col\ [z \in \mathsf{List}(A)] \\ b \in M(\epsilon) \quad c(x, y, z) \in M(\mathsf{cons}(x, y))\ [x \in \mathsf{List}(A), y \in A, z \in M(x)] \end{array}}{\mathsf{El}_{\mathsf{List}}(\epsilon, b, (x, y, z).c) = b \in M(\epsilon)}$$

$$\beta\mathsf{C}_2\text{-List}\ \frac{\begin{array}{l} l \in \mathsf{List}(A) \quad a \in A \quad M(z)\ col\ [z \in \mathsf{List}(A)] \\ b \in M(\epsilon) \quad c(x, y, z) \in M(\mathsf{cons}(x, y))\ [x \in \mathsf{List}(A), y \in A, z \in M(x)] \end{array}}{\mathsf{El}_{\mathsf{List}}(\mathsf{cons}(l, a), b, (x, y, z).c) = c(l, a, \mathsf{El}_{\mathsf{List}}(l, b, (x, y, z).c)) \in M(\mathsf{cons}(l, a))}$$

Quotients

$$\textsf{F-Q}\ \frac{\begin{array}{l} A\ set \quad R(x,y)\ prop_s\ [x \in A, y \in A] \\ R(x,x)\ true\ [x \in A] \\ R(y,x)\ true\ [x,y \in A, p \in R(x,y)] \\ R(x,z)\ true\ [x,y,z \in A, p \in R(x,y), q \in R(y,z)] \end{array}}{A/R\ set}$$

$$\textsf{I-Q}\ \frac{A/R\ set \quad a \in A}{[a] \in A/R}$$

$$\textsf{eq-I-Q}\ \frac{A/R\ set \quad a \in A \quad b \in A \quad R(a,b)\ true}{[a] = [b] \in A/R} \qquad \textsf{eff}\ \frac{[a] = [b] \in A/R}{R(a,b)\ true}$$

$$\textsf{E-Q}\ \frac{\begin{array}{l} M(z)\ col\ [z \in A/R] \\ c \in A/R \quad m(x) \in M([x])\ [x \in A] \\ m(x) = m(y) \in M([x])\ [x \in A, y \in A, p \in R(x,y)] \end{array}}{\mathsf{El_Q}(c, (x).m) \in M(c)}$$

$$\beta\textsf{C-Q}\ \frac{\begin{array}{l} M(z)\ col\ [z \in A/R] \\ a \in A \quad m(x) \in M([x])\ [x \in A] \\ m(x) = m(y) \in M([x])\ [x \in A, y \in A, p \in R(x,y)] \end{array}}{\mathsf{El_Q}([a], (x).m) = m(a) \in M([a])}$$

Congruence rules

$$\textsf{eq-F-Eq}\ \frac{A = C\ col \quad a = b \in A \quad c = d \in C}{\textsf{Eq}(A, a, b) = \textsf{Eq}(C, c, d)\ prop}$$

$$\textsf{eq-F}_\textsf{s}\textsf{-Eq}\ \frac{A = C\ set \quad a = b \in A \quad c = d \in C}{\textsf{Eq}(A, a, b) = \textsf{Eq}(C, c, d)\ prop_s}$$

$$\textsf{eq-F-}\wedge\ \frac{\varphi = \varphi'\ prop \quad \psi = \psi'\ prop}{\varphi \wedge \psi = \varphi' \wedge \psi'\ prop} \qquad \textsf{F}_\textsf{s}\textsf{-}\wedge\ \frac{\varphi = \varphi'\ prop_s \quad \psi = \psi'\ prop_s}{\varphi \wedge \psi = \varphi' \wedge \psi'\ prop_s}$$

$$\textsf{eq-F-}\vee\ \frac{\varphi = \varphi'\ prop \quad \psi = \psi'\ prop}{\varphi \vee \psi = \varphi' \vee \psi'\ prop} \qquad \textsf{F}_\textsf{s}\textsf{-}\vee\ \frac{\varphi = \varphi'\ prop_s \quad \psi = \psi'\ prop_s}{\varphi \vee \psi = \varphi' \vee \psi'\ prop_s}$$

$$\textsf{eq-F-}\Rightarrow\ \frac{\varphi = \varphi'\ prop \quad \psi = \psi'\ prop}{\varphi \Rightarrow \psi = \varphi' \Rightarrow \psi'\ prop} \qquad \textsf{F}_\textsf{s}\textsf{-}\Rightarrow\ \frac{\varphi = \varphi'\ prop_s \quad \psi = \psi'\ prop_s}{\varphi \Rightarrow \psi = \varphi' \Rightarrow \psi'\ prop_s}$$

$$\textsf{eq-F-}\exists\ \frac{A = B\ col \quad \varphi(x) = \psi(x)\ prop\ [x \in A]}{(\exists x \in A)\varphi(x) = (\exists x \in B)\psi(x)\ prop}$$

$$\textsf{eq-F}_\textsf{s}\textsf{-}\exists\ \frac{A = B\ set \quad \varphi(x) = \psi(x)\ prop_s\ [x \in A]}{(\exists x \in A)\varphi(x) = (\exists x \in B)\psi(x)\ prop_s}$$

$$\textsf{eq-F-}\forall\ \frac{A = B\ col \quad \varphi(x) = \psi(x)\ prop\ [x \in A]}{(\forall x \in A)\varphi(x) = (\forall x \in B)\psi(x)\ prop}$$

$$\textsf{eq-F}_\textsf{s}\textsf{-}\forall\ \frac{A = B\ set \quad \varphi(x) = \psi(x)\ prop_s\ [x \in A]}{(\forall x \in A)\varphi(x) = (\forall x \in B)\psi(x)\ prop_s}$$

$$\text{eq-E-N}_0\ \frac{a = b \in \mathsf{N}_0 \quad B(x)\ col\ [x \in \mathsf{N}_0]}{\mathsf{El}_{\mathsf{N}_0}(a) = \mathsf{El}_{\mathsf{N}_0}(b) \in B(a)}$$

$$\text{eq-E-N}_1\ \frac{a = a' \in \mathsf{N}_1 \quad B(x)\ col\ [x \in \mathsf{N}_1] \quad b = b' \in B(\star)}{\mathsf{El}_{\mathsf{N}_1}(a, b) = \mathsf{El}_{\mathsf{N}_1}(a', b') \in B(a)}$$

$$\text{eq-I-}\mathcal{P}(1)\frac{\varphi \Leftrightarrow \psi\ true}{[\varphi] = [\psi] \in \mathcal{P}(1)}$$

$$\text{eq-F-}\Sigma\ \frac{A = B\ col \quad C(x) = D(x)\ col\ [x \in A]}{(\Sigma x \in A)C(x) = (\Sigma x \in B)D(x)\ col}$$

$$\text{eq-F}_{\mathsf{s}}\text{-}\Sigma\ \frac{A = B\ set \quad C(x) = D(x)\ set\ [x \in A]}{(\Sigma x \in A)C(x) = (\Sigma x \in B)D(x)\ set}$$

$$\text{eq-I-}\Sigma\ \frac{a = a' \in A \quad B(x)\ col\ [x \in A] \quad b = b' \in B(a)}{\langle a, b\rangle = \langle a', b'\rangle \in (\Sigma x \in A)B(x)}$$

$$\text{eq-E-}\Sigma\ \frac{\begin{array}{l} c = c' \in (\Sigma x \in A)B(x) \\ M(z)\ col\ [z \in (\Sigma x \in A)B(x)] \quad m(x, y) = m'(x, y) \in M(\langle x, y\rangle)\ [x \in A, y \in B(x)] \end{array}}{\mathsf{El}_\Sigma(c, (x, y).m) = \mathsf{El}_\Sigma(c', (x, y).m') \in M(c)}$$

$$\text{eq-F-}\Pi\ \frac{A = B\ col \quad C(x) = D(x)\ col\ [x \in A]}{(\Pi x \in A)C(x) = (\Pi x \in B)D(x)\ col}$$

$$\text{eq-F-}\Pi\mathcal{P}(1)\ \frac{A = B\ set}{(\Pi x \in A)\mathcal{P}(1) = (\Pi x \in B)\mathcal{P}(1)\ col}$$

$$\text{eq-I-}\Pi\ \frac{t(x) = s(x) \in B(x)\ [x \in A] \quad (\Pi x \in A)B(x)\ col}{(\lambda x \in A)t(x) = (\lambda x \in A)s(x) \in (\Pi x \in A)B(x)}$$

$$\text{eq-E-}\Pi\ \frac{f = g \in (\Pi x \in A)B(x) \quad a = b \in A}{\mathsf{Ap}(f, a) = \mathsf{Ap}(g, b) \in B(a)}$$

$$\text{eq-F-+}\ \frac{A = A'\ set \quad B = B'\ set}{A + B = A' + B'\ set}$$

$$\text{eq-I}_1\text{-+}\ \frac{a = a' \in A \quad A + B\ set}{\mathsf{inl}(a) = \mathsf{inl}(a') \in A + B} \qquad \text{eq-I}_2\text{-+}\ \frac{b = b' \in B \quad A + B\ set}{\mathsf{inr}(b) = \mathsf{inr}(b') \in A + B}$$

$$\text{eq-E-+}\ \frac{\begin{array}{l} c = c' \in A + B \quad M(z)\ col\ [z \in A + B] \\ t(x) = t'(x) \in M(\mathsf{inl}(x))\ [x \in A] \quad s(x) = s'(x) \in M(\mathsf{inr}(x))\ [x \in B] \end{array}}{\mathsf{El}_+(c, (x).t, (x).s) = \mathsf{El}_+(c', (x).t', (x).s') \in M(c)}$$

$$\text{eq-F-List}\ \frac{A = B\ set}{\mathsf{List}(A) = \mathsf{List}(B)\ set}$$

$$\text{eq-I}_2\text{-List}\ \frac{l = l' \in \mathsf{List}(A) \quad a = a' \in A}{\mathsf{cons}(l, a) = \mathsf{cons}(l', a') \in \mathsf{List}(A)}$$

$$\text{eq-E-List}\ \frac{\begin{array}{l} l = l' \in \mathsf{List}(A) \quad M(z)\ col\ [z \in \mathsf{List}(A)] \quad b = b' \in M(\epsilon) \\ c(x, y, z) = c'(x, y, z) \in M(\mathsf{cons}(x, y))\ [x \in \mathsf{List}(A), y \in A, z \in M(x)] \end{array}}{\mathsf{El}_{\mathsf{List}}(l, b, (x, y, z).c) = \mathsf{El}_{\mathsf{List}}(l', b', (x, y, z).c') \in M(l)}$$

$$\text{eq-F-Q}\ \frac{A = B\ set \quad R(x, y) = S(x, y)\ prop_s\ [x \in A, y \in A] \quad A/R\ set}{A/R = B/S\ set}$$

$$\text{eq-I-Q}\ \frac{A/R\ set \quad a \in A \quad b \in A \quad R(a, b)\ true}{[a] = [b] \in A/R}$$

$$\text{eq-E-Q}\ \frac{\begin{array}{l} M(z)\ col\ [z \in A/R] \\ c = c' \in A/R \quad m(x) = m'(x) \in M([x])\ [x \in A] \\ m(x) = m(y) \in M([x])\ [x \in A, y \in A, p \in R(x, y)] \end{array}}{\mathsf{El}_{\mathsf{Q}}(c, (x).m) = \mathsf{El}_{\mathsf{Q}}(c', (x).m') \in M(c)}$$

A.3 Rules for inductive constructors in MLTT

In the following, when we give the rules of a type constructor, we interpret the formation rule's premises as parameters of the constructor; we then take them for granted in the premises of the other constructor's rules.

Rules for dependent wellfounded trees

$$\text{F-DW}\ \frac{\begin{array}{l} A \in \mathsf{U}_0 \\ I(x) \in \mathsf{U}_0\ [x \in A] \\ Br(x,y) \in \mathsf{U}_0\ [x \in A\,, y \in I(x)] \\ ar(x,y) \in Br(x,y) \to A\ [x \in A\,, y \in I(x)] \end{array}}{\mathsf{DW}_{Br,ar} \in A \to \mathsf{U}_0}$$

$$\text{I-DW}\ \frac{a \in A \qquad i \in I(a) \qquad f \in (\Pi z \in Br(a,i))\mathsf{DW}_{Br,ar}(ar(a,i,z))}{\mathsf{dsup}(a,i,f) \in \mathsf{DW}_{Br,ar}(a)}$$

$$\text{E-DW}\ \frac{\begin{array}{l} M(x,w)\ type\ [x \in A\,, w \in \mathsf{DW}_{Br,ar}(x)] \\ d(x,y,h,k) \in M(x, \mathsf{dsup}(x,y,h)) \\ \quad [x \in A, y \in I(x), \\ \qquad h \in (\Pi z \in Br(x,y))\mathsf{DW}_{Br,ar}(ar(x,y,z)), \\ \qquad\quad k \in (\Pi z \in Br(x,y))M(ar(x,y,z), h(z))] \\ a \in A \qquad t \in \mathsf{DW}_{Br,ar}(a) \end{array}}{\mathsf{El}_{\mathsf{DW}}(a,t,(x,y,h,k).d) \in M(a,t)}$$

$$\text{C-DW}\ \frac{\begin{array}{l} M(x,w)\ type\ [x \in A\,, w \in \mathsf{DW}_{Br,ar}(x)] \\ d(x,y,h,k) \in M(x, \mathsf{dsup}(x,y,h)) \\ \quad [x \in A, y \in I(x), \\ \qquad h \in (\Pi z \in Br(x,y))\mathsf{DW}_{Br,ar}(ar(x,y,z)), \\ \qquad\quad k \in (\Pi z \in Br(x,y))M(ar(x,y,z), h(z))] \\ a \in A \quad i \in I(a) \quad f \in (\Pi z \in Br(a,i))\mathsf{DW}_{Br,ar}(ar(a,i,z)) \end{array}}{\mathsf{El}_{\mathsf{DW}}(a, \mathsf{dsup}(a,i,f), d) = d(a,i,f,\lambda z.\mathsf{El}_{\mathsf{DW}}(ar(a,i,z), f(z), d)) \in M(a, \mathsf{dsup}(a,i,f))}$$

Rules for inductive predicates in $\mathbf{ML}_1^\eta$

$$\frac{\begin{array}{l} A \in \mathsf{U}_0 \\ I(x) \in \mathsf{U}_0\ [x \in A] \\ C(x,y) \in A \to \mathsf{U}_0\ [x \in A\,, y \in I(x)] \end{array}}{\mathsf{Ind}_{I,C} \in A \to \mathsf{U}_0}\ \text{F-Ind}$$

$$\frac{a \in A \qquad i \in I(a) \qquad p \in (\Pi x \in A)(C(a,i,x) \to \mathsf{Ind}_{I,C}(x))}{\mathsf{ind}(a,i,p) \in \mathsf{Ind}_{I,C}(a)}\ \text{I-Ind}$$

$$\frac{\begin{array}{l} M(x,w)\ type\ [x \in A\,, w \in \mathsf{Ind}_{I,C}(x)] \\ d(x,y,h,k) \in M(x, \mathsf{ind}(x,y,h)) \\ \quad [x \in A, y \in I(x), \\ \qquad h \in (\Pi z \in A)(C(x,y,z) \to \mathsf{Ind}_{I,C}(x)), \\ \qquad\quad k \in (\Pi z \in A)(\Pi q \in C(x,y,z))M(z, h(z,q))] \\ a \in A \qquad p \in \mathsf{Ind}_{I,C}(a) \end{array}}{\mathsf{El}_{\mathsf{Ind}}(a, p, (x,y,h,k).d) \in M(a,p)}\ \textsf{E-Ind}$$

$$\frac{\begin{array}{l} M(x,w)\ type\ [x \in A\,, w \in \mathsf{Ind}_{I,C}(x)] \\ d(x,y,h,k) \in M(x, \mathsf{ind}(x,y,h)) \\ \quad [x \in A, y \in I(x), \\ \qquad h \in (\Pi z \in A)(C(x,y,z) \to \mathsf{Ind}_{I,C}(x)), \\ \qquad\quad k \in (\Pi z \in A)(\Pi q \in C(x,y,z))M(z, h(z,q))] \\ a \in A \qquad i \in I(a) \qquad p \in (\Pi x \in A)(C(a,i,x) \to \mathsf{Ind}_{I,C}(x)) \end{array}}{\mathsf{El}_{\mathsf{Ind}}(a, \mathsf{ind}(a,i,p), (x,y,h,k).d) = d(a,i,p,\lambda z.\lambda q.\mathsf{El}_{\mathsf{Ind}}(z, p(z,q), d)) \in M(a, \mathsf{ind}(a,i,p))}\ \textsf{C-Ind}$$

Rules for inductive basic covers in $\mathbf{ML}_1^{\eta}$

$$\frac{\begin{array}{l} A \in \mathsf{U}_0 \\ I(x) \in \mathsf{U}_0\ [x \in A] \\ C(x,y) \in A \to \mathsf{U}_0\ [x \in A, y \in I(x)] \\ V \in A \to \mathsf{U}_0 \end{array}}{- \lhd_{I,C} V \in A \to \mathsf{U}_0}\ \textsf{F-}\lhd$$

$$\frac{a \in A \qquad r \in V(a)}{\mathsf{rf}(a,r) \in a \lhd_{I,C} V}\ \mathsf{I}_{\mathsf{rf}}\textsf{-}\lhd$$

$$\frac{a \in A \qquad i \in I(a) \qquad r \in (\Pi x \in A)(C(a,i,x) \to x \lhd_{I,C} V)}{\mathsf{tr}(a,i,r) \in a \lhd_{I,C} V}\ \mathsf{I}_{\mathsf{tr}}\textsf{-}\lhd$$

$$\frac{\begin{array}{l} M(x,w)\ type\ [x \in A, w \in x \lhd_{I,C} V] \\ q_1(x,y) \in M(x, \mathsf{rf}(x,y))\ [x \in A\,, y \in V(a)] \\ q_2(x,y,h,k) \in M(x, \mathsf{tr}(x,y,h)) \\ \quad [x \in A\,, y \in I(x)\,, \\ \quad\quad h \in (\Pi z \in A)(C(x,y,z) \to z \lhd_{I,C} V)\,, \\ \quad\quad\quad k \in (\Pi z \in A)(\Pi q \in C(x,y,z))M(z, h(z,q))] \\ a \in A \qquad p \in a \lhd_{I,C} V \end{array}}{\mathsf{El}_{\lhd}(a, p, (x,y).q_1, (x,y,h,k).q_2) \in M(a,p)}\ \mathsf{E}\text{ - }\lhd$$

$$\frac{\begin{array}{l} M(x,w)\ type\ [x \in A, w \in x \lhd_{I,C} V] \\ q_1(x,y) \in M(x, \mathsf{rf}(x,y))\ [x \in A\,, y \in V(a)] \\ q_2(x,y,h,k) \in M(x, \mathsf{tr}(x,y,h)) \\ \quad [x \in A\,, y \in I(x)\,, \\ \quad\quad h \in (\Pi z \in A)(C(x,y,z) \to z \lhd_{I,C} V)\,, \\ \quad\quad\quad k \in (\Pi z \in A)(\Pi q \in C(x,y,z))M(z, h(z,q))] \\ a \in A \qquad r \in V(a) \end{array}}{\mathsf{El}_{\lhd}(a, \mathsf{rf}(a,r), q_1, q_2) = q_1(a,r) \in M(a, \mathsf{rf}(a,r))}\ \mathsf{C}_{\mathsf{rf}}\text{ - }\lhd$$

$$\frac{\begin{array}{l} M(x,w)\ type\ [x \in A, w \in x \lhd_{I,C} V] \\ q_1(x,y) \in M(x, \mathsf{rf}(x,y))\ [x \in A\,, y \in V(a)] \\ q_2(x,y,h,k) \in M(x, \mathsf{tr}(x,y,h)) \\ \quad [x \in A\,, y \in I(x)\,, \\ \quad\quad h \in (\Pi z \in A)(C(x,y,z) \to z \lhd_{I,C} V)\,, \\ \quad\quad\quad k \in (\Pi z \in A)(\Pi q \in C(x,y,z))M(z, h(z,q))] \\ a \in A \qquad i \in I(a) \qquad r \in (\Pi x \in A)(C(a,i,x) \to x \lhd_{I,C} V) \end{array}}{\mathsf{El}_{\lhd}(a, \mathsf{tr}(a,i,r), q_1, q_2) = q_2(a, i, \lambda z.\lambda q.\mathsf{El}_{\lhd}(z, r(z,q), q_1, q_2)) \in M(a, \mathsf{tr}(a,i,r))}\ \mathsf{C}_{\mathsf{tr}}\text{ - }\lhd$$