Impredicative Observational Equality

LOÏC PUJET, Inria, France
NICOLAS TABAREAU, Inria, France

In dependent type theory, impredicativity is a powerful logical principle that allows the definition of propositions that quantify over arbitrarily large types, potentially resulting in self-referential propositions. Impredicativity can provide a system with increased logical strength and flexibility, but in counterpart it comes with multiple incompatibility results. In particular, Abel and Coquand showed that adding definitional uniqueness of identity proofs (UIP) to the main proof assistants that support impredicative propositions (Coq and Lean) breaks the normalization procedure, and thus the type-checking algorithm. However, it was not known whether this stems from a fundamental incompatibility between UIP and impredicativity or if a more suitable algorithm could decide type-checking for a type theory that supports both. In this paper, we design a theory that handles both UIP and impredicativity by extending the recently introduced observational type theory $\mathsf{TT}^{\mathrm{obs}}$ with an impredicative universe of definitionally proof-irrelevant types, as initially proposed in the seminal work on observational equality of Altenkirch et al. We prove decidability of conversion for the resulting system, that we call $\mathsf{CC}^{\mathrm{obs}}$, by harnessing proof-irrelevance to avoid computing with impredicative proof terms. Additionally, we prove normalization for $\mathsf{CC}^{\mathrm{obs}}$ in plain Martin-Löf type theory, thereby showing that adding proof-irrelevant impredicativity does not increase the computational content of the theory.

CCS Concepts: • **Theory of computation → Type theory**.

Additional Key Words and Phrases: type theory, dependent types, rewriting theory, confluence, termination

ACM Reference Format:
Loïc Pujet and Nicolas Tabareau. 2023. Impredicative Observational Equality. *Proc. ACM Program. Lang.* 7, POPL, Article 74 (January 2023), 26 pages. https://doi.org/10.1145/3571739

1 INTRODUCTION

In dependent type theory, a sort Ω is said to be *impredicative* if it is closed under dependent products over any index type: for all types A and functions $B : A \to \Omega$, the dependent product $\Pi(x : A).\, B\, x$ is in Ω. Impredicativity allows us to define self-referential propositions, which quantify over the type Ω of all propositions and can thus be applied to themselves. In addition to providing a great amount of proof-theoretical strength, impredicativity is a crucial ingredient in many mathematical constructions, such as Tarski's fixed point theorem or lattice theory [Hur et al. 2013]. On the other side of the coin, a *predicative* hierarchy $(\mathcal{U}_i)_{i\in\mathbb{N}}$ requires dependent products to inhabit a higher universe level than their domain and codomain: for all types $A : \mathcal{U}_i$ and functions $B : A \to \mathcal{U}_j$, the dependent product $\Pi(x : A).\, B\, x$ is in $\mathcal{U}_{max(i,j)}$. While predicatives hierarchies are easier to model, as the universes $\mathcal{U}_i$ can be constructed incrementally by induction on the level i, they do not provide all the flexibility of impredicative sorts—whether the mention of levels is explicit (as in Agda) or implicit (as in Coq). Impredicative propositions make for an altogether more comfortable framework, in which less universe levels have to be dealt with.

Authors' addresses: Loïc Pujet, Inria, Gallinette Project-Team, Nantes, France; Nicolas Tabareau, Inria, Gallinette Project-Team, Nantes, France.

2475-1421/2023/1-ART74
https://doi.org/10.1145/3571739

Impredicativity is a prominent feature of the Calculus of Constructions [Coquand and Huet 1988], and by extension of Coq, but it is absent from the standard presentation of Martin-Löf type theory (MLTT) [Martin-Löf 1975] and not available in Agda. This reluctance may be explained by the sheer difficulty of designing models to reason about impredicative theories, and by the numerous incompatibility results, from Hurkens [1995] paradox to the more recent proof by Abel and Coquand [2020] that a naive implementation of uniqueness of identity proofs (UIP) via definitionally proof-irrelevant equalities breaks Coq's normalization algorithm on impredicative propositions.

However, as noted by Abel and Coquand, it is not clear whether that result stems from a fundamental incompatibility between UIP and impredicativity, or if it is an artifact of an inadequate type-checking algorithm. Clarifying this issue seems important given the renewed interest in definitional proof-irrelevance and UIP shown by the community in recent years, especially in the context of observational type theories [Altenkirch et al. 2019; Pujet and Tabareau 2022; Sterling et al. 2019]. In their seminal work on observational type theory, Altenkirch et al. [2007] defined the *observational equality* as a proof-irrelevant proposition that lives in an impredicative sort, which seems to be in tension with the counter-example of Abel and Coquand at first sight. But the computational behavior of the observational equality is different from that of the inductive equality, and Altenkirch et al. [2007] justified their system with an interpretation in a type theory with induction-recursion and proof-irrelevant impredicative propositions, which they conjecture to be normalizing. More recently, Pujet and Tabareau [2022] built a normalization model for a type theory with an observational equality, but their system $\mathsf{TT}^{\mathrm{obs}}$ replaces the impredicative sort of propositions with a predicative hierarchy of proof-irrelevant propositions.

In this paper, we explain how to extend this normalization model to handle an impredicative sort of definitionally proof-irrelevant propositions, by exploiting the fact that it is never necessary to compute with irrelevant proof terms. As a consequence, we can safely extend $\mathsf{TT}^{\mathrm{obs}}$ with impredicativity while preserving definitional UIP and all extensionality principles that come with an observational type theory (such as propositional or function extensionality). The resulting system, which we call $\mathsf{CC}^{\mathrm{obs}}$, should thus be understood as an implementation of the observational type theory of Altenkirch et al. [2007] with a complete proof of normalization and decidability of type-checking. This result has been formalized in Agda using the framework of Abel et al. [2018] for inductive-recursive logical relations.[1]

It may come as a surprise that an inductive-recursive construction which has been designed with predicative theories in mind can be extended to handle an impredicative sort. Indeed, modeling impredicativity usually requires the use of reducibility candidates [Girard 1972], which do not fit well in a proof-irrelevant logical relation. The crux of the proof is to realize that we can get away with remarkably little structure in the interpretation of the impredicative dependent products, so that we can define reducibility for the proof-irrelevant propositions before defining reducibility for types by induction-recursion.

Furthermore, we show that normalization for $\mathsf{CC}^{\mathrm{obs}}$ can be carried in plain Martin-Löf type theory with indexed inductive types, thereby showing that the impredicativity does not contribute to the *computational* power of the proof-relevant fragment at all, despite the increase in *logical* power. We then investigate the possibility of recovering the lost computational power by adding a proof-*relevant* impredicative universe, hoping that the different computational behaviour of $\mathsf{CC}^{\mathrm{obs}}$ would allow us to circumvent Abel and Coquand's argument, but alas, it does not. In fact, a slightly modified argument shows undecidability of type-checking in presence of a proof-relevant impredicative

[1] The formal proof can be found at https://github.com/CoqHott/logrel-mltt/tree/impredicativity-SProp-POPL

universe. Therefore, we exhibit a tradeoff between the comfort of UIP and extensionality principles, and the possibility to extract computational information from impredicative sorts.

Finally, because our normalization proof does not imply logical consistency, we define a model of $\mathrm{CC}^{\mathrm{obs}}$ in ZF Set theory. This shows that the computational content of $\mathrm{CC}^{\mathrm{obs}}$ can be studied without impredicativity in the metatheory, while the study of the logical content of $\mathrm{CC}^{\mathrm{obs}}$ fundamentally requires an impredicative metatheory.

Plan of the paper. In Section 2, we present the syntax and typing rules of $\mathrm{CC}^{\mathrm{obs}}$, focusing on the differences with respect to $\mathsf{TT}^{\mathrm{obs}}$. Then, we develop the logical relation framework that shows normalization and decidability of conversion for $\mathrm{CC}^{\mathrm{obs}}$ (Section 3). In Section 4, we show that $\mathrm{CC}^{\mathrm{obs}}$ cannot express more integer functions as closed terms of type $\mathbb{N} \to \mathbb{N}$ than MLTT, by simplifying the logical relation framework to avoid the use of induction-recursion. Finally, we show logical consistency of $\mathrm{CC}^{\mathrm{obs}}$ in Section 5 and adapt the undecidability argument in presence of a proof-relevant impredicative universe of Abel and Coquand [2020] to our setting in Section 6.

2 AN IMPREDICATIVE TYPE THEORY WITH OBSERVATIONAL EQUALITY

$\mathrm{CC}^{\mathrm{obs}}$ has both a predicative hierarchy of universes, that is written as $\mathcal{U}_i$ (where i is a natural number) and works just like the usual universe hierarchy of Martin-Löf type theory, and an impredicative universe Ω of *strict propositions*. Strict propositions are definitionally proof-irrelevant types, which means that any two inhabitants of a strict proposition $A : \Omega$ are always convertible.

The impredicativity of Ω departs significantly from the predicative hierarchy of strict propositions used in Gilbert et al. [2019], used for $\mathsf{TT}^{\mathrm{obs}}$ and implemented in Agda. Instead, it is closer to the proof-irrelevant impredicative universe SProp of Coq. With this impredicative universe of propositions, $\mathrm{CC}^{\mathrm{obs}}$ appears somewhat more natural than $\mathsf{TT}^{\mathrm{obs}}$, as the observational equality can remain in Ω even when the type on which equality is defined lives higher up in the universe hierarchy—which means we don't need cumulative dependent products anymore.

Apart from this fundamental difference, $\mathrm{CC}^{\mathrm{obs}}$ supports the same constructions as $\mathsf{TT}^{\mathrm{obs}}$. It has a notion of observational equality that can be formed on any proof-relevant type, and computes by case analysis on the type. The observational equality is an element of Ω, but it can be eliminated into $\mathcal{U}_i$ by means of a cast operator between two equal types, which computes separately on each pair of types. The non-parametric computation rules for equality and cast allow us to get function extensionality and proposition extensionality by definition, in addition to the definitional principle of UIP that comes with proof-irrelevant equality types.

2.1 The Syntax of $\mathrm{CC}^{\mathrm{obs}}$

The syntax of the sorts, contexts, terms and types of $\mathrm{CC}^{\mathrm{obs}}$ is specified in Fig. 1. It is largely similar to the syntax of $\mathsf{TT}^{\mathrm{obs}}$, with the one difference of having only one impredicative universe Ω of strict propositions. The theory features dependent function types (noted $\Pi^{s,s'}(x : A).\, B$, or simply $\Pi(x : A).\, B$ when the sorts can be inferred from the context) with η-equality, natural numbers, as well as proof-irrelevant dependent conjunctions (noted $(x : A) \,\&\, B$), a proof-irrelevant empty type (noted $\bot$) and unit type (noted $\top$). When B does not depend on A, we write $A \to B$ instead of $\Pi(x : A).\, B$, and $A \wedge B$ instead of $(x : A) \,\&\, B$. The capture-avoiding substitution of a variable x in a term A by the term t is noted $B[x := t]$. As with $\mathsf{TT}^{\mathrm{obs}}$, $\mathrm{CC}^{\mathrm{obs}}$ can be extended with more primitives such as dependent sums, quotients or general inductive types, but we do not treat them in this paper to better focus on the treatment of impredicativity in the theory.

In addition to the usual syntax of MLTT with proof irrelevant types, there are two new ingredients that were introduced with $\mathsf{TT}^{\mathrm{obs}}$, and are present in $\mathrm{CC}^{\mathrm{obs}}$: the first one is a proof-irrelevant observational equality type $t \sim_A u$ that encodes equality of t and u at type A, and the second is an

i, j	$\in$	$\mathbb{N}$	Universe levels
s	$::=$	$\mathcal{U}_i \mid \Omega$	Universes
Γ, Δ	$::=$	$\bullet \mid \Gamma, x : A : s$	Contexts
t, u, m, n, e, A, B	$::=$	$x \mid s$	Variables and Universes
	$\mid$	$\lambda(x : A).\, t \mid t\ u \mid \Pi^{s,s'}(x : A).\, B$	Dependent products
	$\mid$	$0 \mid \mathrm{S}\ t \mid \mathbb{N}\text{-elim}(P, t, u, n) \mid \mathbb{N}$	Natural numbers
	$\mid$	$\langle t, u\rangle \mid \mathrm{fst}(t) \mid \mathrm{snd}(t) \mid (x : A)\ \&\ B$	Dependent conjunction
	$\mid$	$\bot\text{-elim}(A, t) \mid \bot$	Empty type
	$\mid$	$* \mid \top$	Singleton type
	$\mid$	$t \sim_A u \mid \mathrm{refl}(t) \mid \mathrm{transp}(A, t, B, u, t', e)$	Observational equality
	$\mid$	$\mathrm{cast}(A, B, e, t) \mid \mathrm{castrefl}(A, t)$	Type cast

Fig. 1. Syntax of $\mathrm{CC}^{\mathrm{obs}}$

operator that casts a term t from a type A to another type B given a proof of equality e between A and B, which is noted $\mathrm{cast}(A, B, e, t)$.

Observational equality is quite different from the usual Martin-Löf Identity Type, which is an inductive type that computes via the J eliminator. Instead, observational equality $t \sim_A u$ is better thought of as an eliminator that reduces by case analysis on the weak head normal form of A, much like the composition operation from Cubical Type Theory [Cohen et al. 2015].

2.2 The Typing Rules of $\mathrm{CC}^{\mathrm{obs}}$

The typing rules of $\mathrm{CC}^{\mathrm{obs}}$ are presented in Fig. 2. They are based on five judgments: $\vdash \Gamma$ (well-formed contexts), $\Gamma \vdash t : A : s$ (well-typed terms), $\Gamma \vdash A : s$ (well-formed types) $\Gamma \vdash t \equiv u : A : s$ (convertibility of terms), and $\Gamma \vdash A \equiv B : s$ (convertibility of types). In all the judgments, s denotes either $\mathcal{U}_i$ or Ω. In addition to the typing judgments, we describe a weak-head reduction strategy for proof relevant terms that we will use for our normalization procedure, which is written $\Gamma \vdash t \Rightarrow u : A$. Remark that because our weak-head reduction only applies to terms of a proof-relevant type, we do not need sort annotations.

Since every universe has a type, the well-formedness judgments for types $\Gamma \vdash A : s$ (and convertibility judgments of types) can be seen as special cases of typing judgements for terms $\Gamma \vdash A : s : s'$ for a suitable s', but we believe that keeping this distinction helps the presentation.

We use the Pure Type System presentation [Barendregt 1993] to factorize the impredicative and predicative rules for universes and dependent function types.

Generic rules and universes. Rules Ctx-Nil, Ctx-Cons and Var describe the usual formation of contexts and typing of variables. The only remarkable thing here is that variables of the context are equipped with both a type and its sort.

Rule conv stipulates that type checking is done modulo conversion of types. The formation rule for universes (Rule Univ) states that both $\mathcal{U}_i$ and Ω are relevant types in a higher universe, as described by the relations

$$\mathcal{A}(\mathcal{U}_i, \mathcal{U}_j) \quad := \quad j = i + 1 \qquad\qquad \mathcal{A}(\Omega, \mathcal{U}_i) \quad := \quad i = 0.$$

The reader should pay attention to the fact that the universe of proof-irrelevant types Ω is a proof-*relevant* type. Even though the inhabitants in Ω are codes for proof-irrelevant types, the codes themselves are computationally relevant.

Dependent products. We allow the formation of dependent products with a domain and a codomain that have different sorts. The resulting type has the same relevance as the codomain, and in the

$$\textsc{Ctx-Nil}\ \frac{}{\vdash \bullet} \qquad \textsc{Ctx-Cons}\ \frac{\vdash \Gamma \qquad \Gamma \vdash A : s}{\vdash \Gamma, x : A : s} \qquad \textsc{Var}\ \frac{\vdash \Gamma \qquad x : A : s \in \Gamma}{\Gamma \vdash x : A : s} \qquad \textsc{conv}\ \frac{\Gamma \vdash t : A : s \qquad \Gamma \vdash A \equiv B : s}{\Gamma \vdash t : B : s}$$

$$\textsc{Univ}\ \frac{\vdash \Gamma}{\Gamma \vdash s : s'}\,\mathcal{A}(s,s') \qquad \Pi\textsc{-Form}\ \frac{\Gamma \vdash A : s \qquad \Gamma, x : A : s \vdash B : s'}{\Gamma \vdash \Pi^{s,s'}(x : A).\, B : \mathcal{R}(s,s')} \qquad \&\textsc{-Form}\ \frac{\Gamma \vdash A : \Omega \qquad \Gamma, x : A : \Omega \vdash B : \Omega}{\Gamma \vdash (x : A)\ \&\ B : \Omega}$$

$$\textsc{Fun}\ \frac{\Gamma \vdash A : s \qquad \Gamma, x : A : s \vdash t : B : s'}{\Gamma \vdash \lambda(x : A).\, t : \Pi^{s,s'}(x : A).\, B : \mathcal{R}(s,s')}$$

$$\textsc{App}\ \frac{\Gamma \vdash A : s \qquad \Gamma, x : A : s \vdash B : s' \qquad \Gamma \vdash t : \Pi^{s,s'}(x : A).\, B : \mathcal{R}(s,s') \qquad \Gamma \vdash u : A : s}{\Gamma \vdash t\ u : B[x := u] : s'}$$

$$\textsc{Pair}\ \frac{\Gamma \vdash A : \Omega \qquad \Gamma, x : A : \Omega \vdash B : \Omega \qquad \Gamma \vdash t : A : \Omega \qquad \Gamma \vdash u : B[x := t] : \Omega}{\Gamma \vdash \langle t, u \rangle : (x : A)\ \&\ B : \Omega}$$

$$\textsc{Fst}\ \frac{\Gamma \vdash A : \Omega \qquad \Gamma, x : A : \Omega \vdash B : \Omega \qquad \Gamma \vdash t : (x : A)\ \&\ B : \Omega}{\Gamma \vdash \mathrm{fst}(t) : A : \Omega} \qquad \textsc{Snd}\ \frac{\text{Same premises as } \textsc{Fst}}{\Gamma \vdash \mathrm{snd}(t) : B[x := \mathrm{fst}(t)] : \Omega}$$

$$\mathbb{N}\textsc{-Form}\ \frac{\vdash \Gamma}{\Gamma \vdash \mathbb{N} : \mathcal{U}_0} \qquad \textsc{Zero}\ \frac{\vdash \Gamma}{\Gamma \vdash 0 : \mathbb{N} : \mathcal{U}_0} \qquad \textsc{Suc}\ \frac{\Gamma \vdash n : \mathbb{N} : \mathcal{U}_0}{\Gamma \vdash \mathrm{S}\ n : \mathbb{N} : \mathcal{U}_0}$$

$$\mathbb{N}\textsc{-Elim}\ \frac{\Gamma, m : \mathbb{N} \vdash A : s \qquad \Gamma \vdash t_0 : A[m := 0] : s \qquad \Gamma \vdash t_{\mathrm{S}} : \Pi(n : \mathbb{N}).\, A[m := n] \to A[m := \mathrm{S}\ n] : s \qquad \Gamma \vdash n : \mathbb{N} : \mathcal{U}_0}{\Gamma \vdash \mathbb{N}\text{-elim}(A, t_0, t_{\mathrm{S}}, n) : A[m := n] : s}$$

$$\bot\textsc{-Form}\ \frac{\vdash \Gamma}{\Gamma \vdash \bot : \Omega} \qquad \bot\textsc{-Elim}\ \frac{\Gamma \vdash A : s \qquad \Gamma \vdash t : \bot : \Omega}{\Gamma \vdash \bot\text{-elim}(A, t) : A : s} \qquad \top\textsc{-Form}\ \frac{\vdash \Gamma}{\Gamma \vdash \top : \Omega} \qquad \top\textsc{-Intro}\ \frac{\vdash \Gamma}{\Gamma \vdash * : \top : \Omega}$$

$$\textsc{Eq-Form}\ \frac{\Gamma \vdash A : \mathcal{U}_i \qquad \Gamma \vdash t : A : \mathcal{U}_i \qquad \Gamma \vdash u : A : \mathcal{U}_i}{\Gamma \vdash t \sim_A u : \Omega} \qquad \textsc{Refl}\ \frac{\Gamma \vdash A : \mathcal{U}_i \qquad \Gamma \vdash t : A : \mathcal{U}_i}{\Gamma \vdash \mathrm{refl}(t) : t \sim_A t : \Omega}$$

$$\textsc{Cast}\ \frac{\Gamma \vdash A, B : s \qquad \Gamma \vdash e : A \sim_s B : \Omega \qquad \Gamma \vdash t : A : s}{\Gamma \vdash \mathrm{cast}(A, B, e, t) : B : s} \qquad \textsc{Cast-Refl}\ \frac{\Gamma \vdash A : s \qquad \Gamma \vdash t : A : s}{\Gamma \vdash \mathrm{castrefl}(A, t) : t \sim_A \mathrm{cast}(A, A, \mathrm{refl}(t), t)}$$

$$\textsc{Transport-}\Omega\ \frac{\Gamma \vdash A : \mathcal{U}_i \qquad \Gamma \vdash t, t' : A : \mathcal{U}_i \qquad \Gamma, y : A : \mathcal{U}_i \vdash B : \Omega \qquad \Gamma \vdash u : B[y := t] : \Omega \qquad \Gamma \vdash e : t \sim_A t' : \Omega}{\Gamma \vdash \mathrm{transp}(A, t, B, u, t', e) : B[y := t'] : \Omega}$$

Fig. 2. $\mathrm{CC}^{\mathrm{obs}}$ Typing Rules

proof-relevant case we ask the universe level of the dependent product to be higher than both the level of the domain and that of the codomain. (Rule Π-Form). This is made formal by using the function $\mathcal{R}(\_,\_)$ defined as

$$\mathcal{R}(s,\Omega) \;:=\; \Omega \qquad \mathcal{R}(\Omega,\mathcal{U}_i) \;:=\; \mathcal{U}_i \qquad \mathcal{R}(\mathcal{U}_i,\mathcal{U}_j) \;:=\; \mathcal{U}_{max(i,j)}.$$

Rules Fun and App are the rules for the introduction of a λ-abstraction and application.

Dependent conjunction, Empty and Unit types. $\mathsf{CC}^{\mathsf{obs}}$ features proof-irrelevant dependent conjunctions (&-Form), an empty proposition (Rule ⊥-Form) and the true proposition $\top$ (Rule ⊤-Form). Those type constructors are helpful as they appear in the computation rules of the observational equality—in particular, the dependent conjunction is used to specify the observational equality between two dependent function types, as a dependent pair of equalities between the domains and the codomains—but we could alternatively define them *via* impredicative encodings.

We give a negative presentation of dependent conjunctions with one introduction rule (Rule Pair) and two projections (Rules Fst and Snd). Note that because of proof-irrelevance, the negative and positive presentations are completely equivalent, and we do not need any rule to account for the computational behavior of projections, nor η-equality. Besides its formation rule (Rule ⊥-Form), the empty type comes with an elimination principle (Rule ⊥-Elim) that can eliminate it into both proof-irrelevant and proof-relevant types. The unit type on the other hand only has a formation rule (Rule ⊤-Form) and a constructor $*$ (Rule ⊤-Intro). It does not require an eliminator, since the usual one can be derived from proof irrelevance.

Natural numbers. All inductive types and quotient types defined in $\mathsf{TT}^{\mathsf{obs}}$ could be added to the core of $\mathsf{CC}^{\mathsf{obs}}$. Here, we only treat the type of natural numbers, as it strikes a good balance between simplicity and illustrating how we deal with recursive constructors and large elimination. The type $\mathbb{N}$ comes with a formation rule $\mathbb{N}$-Form, constructors 0 (Rule Zero) and S (Rule Suc) and an elimination/induction principle given by Rule $\mathbb{N}$-Elim.

Equality and Type Casts. A central feature of $\mathsf{CC}^{\mathsf{obs}}$ is that every proof-relevant type comes equipped with a proof-irrelevant equality type, noted $t \sim_A u$ (Rule Eq-Form) and a canonical way to inhabit it, with the reflexivity term $\mathsf{refl}(t)$ (Rule Refl). Proof-irrelevant types are not equipped with a propositional equality, as any two terms would always be propositionally equal by reflexivity.

For this equality type to be of any use, we need to be able to eliminate it as well. The usual J eliminator from MLTT, called transp, is restricted to proof-irrelevant predicates (Rule Transport-Ω). The elimination of equality in proof-relevant predicates is split into two parts: on the one hand, we use the cast primitive to handle transport between two propositionally equal types (Rule Cast), and on the other hand we can obtain a proof of equality between $B\ x$ and $B\ y$ by using transp to show that function application preserves the equality. Concretely, we can define

$$\begin{aligned} \mathsf{ap}\ B\ e &= \mathsf{transp}(A, x, \lambda(t : A).\, B\ x \sim B\ t, \mathsf{refl}(B\ x), y, e) \\ J(A, x, B, y, e) &= \mathsf{cast}(B\ x, B\ y, \mathsf{ap}\ B\ e, x). \end{aligned}$$

The term $\mathsf{cast}(A, B, e, t)$ is an eliminator that reduces by case analysis on the head constructors of the type A and B. This will be explained in more detail in the next section. When applied to reflexivity, cast does not compute as the identity function, but this equality is propositionally admissible as witnessed by $\mathsf{castrefl}$ (Rule Cast-Refl).

$$\textsc{Refl}\ \frac{\Gamma \vdash t : A : \mathcal{U}_i}{\Gamma \vdash t \equiv t : A : \mathcal{U}_i} \qquad \textsc{Sym}\ \frac{\Gamma \vdash t \equiv u : A : \mathcal{U}_i}{\Gamma \vdash u \equiv t : A : \mathcal{U}_i} \qquad \textsc{Trans}\ \frac{\Gamma \vdash t \equiv t' : A : \mathcal{U}_i \qquad \Gamma \vdash t' \equiv u : A : \mathcal{U}_i}{\Gamma \vdash t \equiv u : A : \mathcal{U}_i} \qquad \textsc{Red-Conv}\ \frac{\Gamma \vdash t \Rightarrow u : A}{\Gamma \vdash t \equiv u : A : \mathcal{U}_i}$$

$$\eta\textsc{-Eq}\ \frac{\Gamma \vdash A : s \qquad \Gamma \vdash t, u : \Pi^{s,\mathcal{U}_i}(x : A).\, B : \mathcal{R}(s, \mathcal{U}_i) \qquad \Gamma, x : A : s \vdash t\ x \equiv u\ x : B : \mathcal{U}_i}{\Gamma \vdash t \equiv u : \Pi^{s,\mathcal{U}_i}(x : A).\, B : \mathcal{R}(s, \mathcal{U}_i)} \qquad \textsc{Proof-Irr}\ \frac{\Gamma \vdash t, u : A : \Omega}{\Gamma \vdash t \equiv u : A : \Omega}$$

Fig. 3. $\mathrm{CC}^{\mathrm{obs}}$ Conversion Rules (except congruence rules)

$$\beta\textsc{-red}\ \frac{\Gamma \vdash A : s \qquad \Gamma, x : A : s \vdash B : \mathcal{U}_i \qquad \Gamma, x : A \vdash t : B : \mathcal{U}_i \qquad \Gamma \vdash u : A : s}{\Gamma \vdash (\lambda(x : A).\, t)\ u \Rightarrow t[x := u] : B[x := u]} \qquad \textsc{Conv-Red}\ \frac{\Gamma \vdash t \Rightarrow u : A \qquad \Gamma \vdash A \equiv B : \mathcal{U}_i}{\Gamma \vdash t \Rightarrow u : B}$$

$$\mathbb{N}\textsc{-Elim-Zero}\ \frac{\Gamma, m : \mathbb{N} \vdash A : s \qquad \Gamma \vdash t_0 : A[m := 0] : s \qquad \Gamma \vdash t_{\mathrm{S}} : \Pi(n : \mathbb{N}).\, A[m := n] \to A[m := \mathrm{S}\ n] : s}{\Gamma \vdash \mathbb{N}\text{–elim}(A, t_0, t_{\mathrm{S}}, 0) \Rightarrow t_0 : A[m := 0]}$$

$$\mathbb{N}\textsc{-Elim-Suc}\ \frac{\Gamma, m : \mathbb{N} \vdash A : s \qquad \Gamma \vdash t_0 : A[m := 0] : s \qquad \Gamma \vdash t_{\mathrm{S}} : \Pi(n : \mathbb{N}).\, A[m := n] \to A[m := \mathrm{S}\ n] : s \qquad \Gamma \vdash n : \mathbb{N}}{\Gamma \vdash \mathbb{N}\text{–elim}(A, t_0, t_{\mathrm{S}}, \mathrm{S}\ n) \Rightarrow_{\mathrm{S}} \mathbb{N}\text{–elim}(A, t_0, t_{\mathrm{S}}, n) : A[m := \mathrm{S}\ n]}$$

$$\textsc{Eq-Fun}\ \frac{\Gamma \vdash A : s \qquad \Gamma, x : A : s \vdash B : \mathcal{U}_i \qquad \Gamma \vdash f, g : \Pi(x : A).\, B : \mathcal{U}_i}{\Gamma \vdash f \sim_{\Pi AB} g \Rightarrow \Pi(x : A).\, f\ x \sim_B g\ x : \Omega} \qquad \textsc{Eq-}\Omega\ \frac{\Gamma \vdash A : \Omega \qquad \Gamma \vdash B : \Omega}{\Gamma \vdash A \sim_\Omega B \Rightarrow (A \to B) \wedge (B \to A) : \Omega}$$

$$\textsc{Eq-Univ}\ \frac{\vdash \Gamma \qquad A \in \{\mathbb{N}, s\}}{\Gamma \vdash A \sim_{s'} A \Rightarrow \top : \Omega}\,\mathcal{A}(s,s') \qquad \textsc{Eq-Univ-}\neq\ \frac{\vdash \Gamma \qquad A, B \in \{\mathbb{N}, \Pi A.B, s\} \qquad \mathrm{hd}\, A \neq \mathrm{hd}\, B}{\Gamma \vdash A \sim_{s'} B \Rightarrow \bot : \Omega}\,\mathcal{A}(s,s')$$

$$\textsc{Eq-}\Pi\ \frac{\Gamma \vdash A : s \qquad \Gamma \vdash A' : s \qquad \Gamma, x : A \vdash B : s' \qquad \Gamma, x : A' \vdash B' : s'}{\Gamma \vdash \begin{array}{l}\Pi(x : A).\, B \sim_{\mathcal{R}(s,s')} \Pi(x : A').\, B' \Rightarrow \\ (e : A' \sim_s A)\ \&\ \Pi(a' : A').\, B[x := \mathrm{cast}(A', A, e, a')] \sim_{s'} B'[x := a']\end{array} : \Omega}$$

$$\textsc{Eq-zero}\ \frac{\vdash \Gamma}{\Gamma \vdash 0 \sim_{\mathbb{N}} 0 \Rightarrow \top : \Omega} \qquad \textsc{Eq-Suc}\ \frac{\Gamma \vdash n : \mathbb{N} : \mathcal{U}_0 \qquad \Gamma \vdash m : \mathbb{N} : \mathcal{U}_0}{\Gamma \vdash \mathrm{S}\ m \sim_{\mathbb{N}} \mathrm{S}\ n \Rightarrow m \sim_{\mathbb{N}} n : \Omega} \qquad \textsc{Eq-zero-suc}\ \frac{\Gamma \vdash n : \mathbb{N} : \mathcal{U}_0}{\Gamma \vdash 0 \sim_{\mathbb{N}} \mathrm{S}\ n \Rightarrow \bot : \Omega} \qquad \textsc{Eq-suc-zero}\ \frac{\Gamma \vdash n : \mathbb{N} : \mathcal{U}_0}{\Gamma \vdash \mathrm{S}\ n \sim_{\mathbb{N}} 0 \Rightarrow \bot : \Omega}$$

$$\textsc{Cast-Zero}\ \frac{\Gamma \vdash e : \mathbb{N} \sim_{\mathcal{U}_0} \mathbb{N} : \Omega}{\Gamma \vdash \mathrm{cast}(\mathbb{N}, \mathbb{N}, e, 0) \Rightarrow 0 : \mathbb{N}} \qquad \textsc{Cast-Suc}\ \frac{\Gamma \vdash e : \mathbb{N} \sim_{\mathcal{U}_0} \mathbb{N} : \Omega \qquad \Gamma \vdash n : \mathbb{N} : \mathcal{U}_0}{\Gamma \vdash \mathrm{cast}(\mathbb{N}, \mathbb{N}, e, \mathrm{S}\ n) \Rightarrow \mathrm{S}\ \mathrm{cast}(\mathbb{N}, \mathbb{N}, e, n) : \mathbb{N}} \qquad \textsc{Cast-Univ}\ \frac{\Gamma \vdash e : s \sim_{s'} s \qquad \Gamma \vdash A : s}{\Gamma \vdash \mathrm{cast}(s, s, e, A) \Rightarrow A : s}$$

$$\textsc{Cast-}\Pi\ \frac{\begin{array}{c}\Gamma \vdash A, A' : s \qquad \Gamma, x : A \vdash B : s' \qquad \Gamma, x : A' \vdash B' : s' \\ \Gamma \vdash e : \Pi(x : A).\, B \sim_{\mathcal{U}_i} \Pi(x : A').\, B' : \Omega \qquad \Gamma \vdash f : \Pi(x : A).\, B \qquad a := \mathrm{cast}(A', A, \mathrm{fst}(e), a')\end{array}}{\Gamma \vdash \begin{array}{l}\mathrm{cast}(\Pi(x : A).\, B, \Pi(x : A').\, B', e, f) \Rightarrow \\ \lambda(a' : A').\, \mathrm{cast}(B[x := a], B'[x := a'], \mathrm{snd}(e)\ a', f\ a)\end{array} : \Pi(x : A').\, B'}$$

Fig. 4. $\mathrm{CC}^{\mathrm{obs}}$ Reduction Rules (except substitutions)

$$
\begin{array}{llll}
\text{whnf} & w & ::= & N \mid \Pi(x : A).\, B \mid s \mid (x : A)\ \&\ B \mid \mathbb{N} \mid \bot \mid \top \mid \lambda(x : A).\, t \mid 0 \mid \mathrm{S}\ n \\
\text{neutral} & N & ::= & x \mid N\ t \mid \bot\text{-elim}(A, e) \mid \mathbb{N}\text{-elim}(P, t, u, N) \\
& & & \mid t \sim_N u \mid N \sim_{\mathbb{N}} m \mid 0 \sim_{\mathbb{N}} N \mid \mathrm{S}\ m \sim_{\mathbb{N}} N \\
& & & \mid N \sim_{\mathcal{U}_i} B \mid \mathbb{N} \sim_{\mathcal{U}_i} N \mid \Pi(x : A).\, B \sim_{\mathcal{U}_i} N \mid \mathrm{cast}(\mathbb{N}, \mathbb{N}, e, N) \\
& & & \mid \mathrm{cast}(N, B, e, t) \mid \mathrm{cast}(\mathbb{N}, N, e, t) \mid \mathrm{cast}(\Pi(x : A).\, B, N, e, t) \\
& & & \mid \mathrm{cast}(w, w', e, t) \qquad (\text{where } w, w' \in \{\mathbb{N}, \Pi A.B, s_i\}, \mathrm{hd}\, w \neq \mathrm{hd}\, w')
\end{array}
$$

Fig. 5. Weak-head normal and neutral forms

2.3 Conversion and Reduction to Weak-Head Normal Forms

Conversion (Fig. 3) subsumes reduction (Red-Conv), η-equality of functions (Rule η-Eq), and proof-irrelevance (Rule Proof-Irr). It is also closed under reflexivity, symmetry, transitivity and congruence rules (e.g. if $t \equiv t'$ and $u \equiv u'$ then $t\ u \equiv t'\ u'$).

Fig. 4 describes the weak-head reduction strategy for $\mathrm{CC}^{\mathrm{obs}}$ which is very similar to the weak-head reduction for $\mathrm{TT}^{\mathrm{obs}}$, with the exception that the impredicativity of Ω removes the need for cumulativity. Since our reduction relation is typed, we need a rule for type conversion (Rule Conv-Red). Rule β-red is the standard β-reduction for proof-relevant function applications, and rules $\mathbb{N}$-Elim-Zero and $\mathbb{N}$-Elim-Suc are the usual computation rules for the elimination principle of natural numbers. Then we have nine rules describing the behaviour of equality for each pair of type constructors. First, rule Eq-Fun bakes function extensionality for the dependent products in our system, and rule Eq-Ω provides propositional extensionality for proof-irrelevant propositions. Rule Eq-Univ gives us reflexivity for the type of natural numbers and universes. Dually, rule Eq-Univ-$\neq$ corresponds to anti-diagonal cases where we can discriminate over the type constructors and know that the equality is not possible (this rule is not strictly necessary to get a well-behaved system, but it is a natural addition to the type theory). Rule Eq-Π states that an equality between two dependent products consists in a pair of an equality between the domains and an equality between the codomains up to casting of the argument in the domain. Note that the equality on the domain is flipped so that casting is more natural for the codomain, as it is done in [Altenkirch et al. 2007]. The four next rules deal with equality of natural numbers in the expected way. Finally, there are four rules describing the behaviour of the cast operator between two matching type constructors. The most interesting case is a cast between two dependent functions, which reduces to a function that casts its argument, applies the original function and then casts back the result. This rule performs an η-expansion on the fly, which is not an issue as η-equality is valid in $\mathrm{CC}^{\mathrm{obs}}$.

A notion that will play a central role in our normalization procedure is that of a *weak-head normal form* (whnf), which corresponds to a relevant term that cannot be reduced further (Fig. 5). Weak-head normal forms are either terms with a constructor in head position, or *neutral terms* stuck on a variable or an elimination of a proof of $\bot$. In other words, neutral terms are weak-head normal forms that should not exist in an empty context. In $\mathrm{CC}^{\mathrm{obs}}$, inhabitants of a proof-irrelevant type are never considered as whnf, as there is no notion of reduction of proof-irrelevant terms.

2.4 $\mathrm{CC}^{\mathrm{obs}}$ in Practice

The combination of impredicative propositions with the observational equality provide several important reasoning principles for mathematics.

Large elimination of propositions. The universe Ω does not support large elimination of sub-singleton inductive propositions, contrary to the universe $\mathbb{P}\mathrm{rop}$ of Coq, but there are two important propositions that can be eliminated into $\mathcal{U}$. First, the eliminator of the empty proposition $\bot$ applies

$\Gamma \Vdash_\ell A : s$	A is a reducible type of sort s in context Γ
$\Gamma \Vdash_\ell A \equiv B : s$	A and B are reducibly equal types of sort s in context Γ
$\Gamma \Vdash_\ell t : A : s$	t is a reducible term of type A, which has sort s in context Γ
$\Gamma \Vdash_\ell t \equiv u : A : s$	t and u are reducibly equal terms of type A, which has sort s in context Γ

Fig. 6. The four judgments of the logical relation

indiscriminately to all types. As a consequence, we can reason in Ω to show that a case of a pattern-matching is not accessible inside a proof-relevant computation, as described in detail by Gilbert et al. [2019]. The other proposition that can be eliminated into $\mathcal{U}$ is the observational equality, using the cast operator. From it, we can define the J eliminator of Martin-Löf identity type as explained in Section 2.2, and thus recover Leibniz's principle of identity of indiscernibles.

Extensionality principles. The observational equality of $\mathrm{CC}^{\mathrm{obs}}$ validates three extensionality principles that are commonplace in mathematics but not available in MLTT. First, it satisfies the principle of uniqueness of identity proofs (UIP), since any two proofs of the same equality are convertible by virtue of being inhabitants of a proof-irrelevant type. Second, the observational equality provides us with the principle of function extensionality, thanks to the computation rule Eq-Π that specifies the equality between two dependent functions. Function extensionality quickly becomes essential when reasoning on higher-order concepts, such as monads and categories. Last but not least, $\mathrm{CC}^{\mathrm{obs}}$ validates the principle of propositional extensionality thanks to the rule Eq-Ω.

Impredicativity of Ω. Impredicativity is the main new feature compared to the system $\mathsf{TT}^{\mathrm{obs}}$ of [Pujet and Tabareau 2022]. As mentioned in the introduction, having impredicativity allows us to use Ω as a complete lattice of truth values, and thus to use impredicative encodings and fixed point theorems. Defining properties as least fixed points proved to be an important tool in the design of compositional coinductive proofs [Hur et al. 2013]. On a more down-to-earth level, impredicativity also simplifies the universe level management, which comes in handy in many definitions. For instance the partiality monad

$$\mathrm{Partial}\ (A : \mathcal{U}_i) := \Sigma(P : \Omega)\ .\ P \to A : \mathcal{U}_i$$

can be defined only for a single universe Ω of proposition, instead of having a family of such partiality monads indexed by the level of the proposition to be used.

3 A LOGICAL RELATION WITH AN IMPREDICATIVE UNIVERSE

In this section, we define a logical relation by following the strategy of Abel et al. [2018] to show both the normalization theorem, which states that any well-typed term with a proof-relevant type can be reduced down to a weak-head normal form by repeatedly applying the weak-head reduction strategy, and decidability of the conversion of $\mathrm{CC}^{\mathrm{obs}}$.

Normalization of type theories is a subtle matter: if one tries to prove it by naive induction on the typing derivations, one quickly gets stuck on the case of the application rule because we cannot deduce normalization of $t\ u$ from normalization of t and u. The so-called "logical relation" is a standard technique to generalize the induction hypothesis, so that the proof by induction on the typing derivation goes through. Concretely, the logical relation is a complex inductive-recursive family of predicates that collect information on types and terms in order to thoroughly characterizes their behaviour with respect to reduction, in a way that directly implies normalization. Types and terms that satisfy the logical relation are called *reducible*, and our aim is to show the *fundamental lemma*, which states that every well-typed term is reducible.

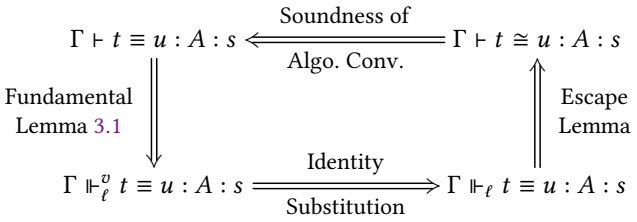

Fig. 7. Outline of the proof

The logical relation consists of four predicates which mirror the four typing judgments of $\mathrm{CC}^{\mathrm{obs}}$ (cf Fig. 6), defined on a type-by-type basis. For MLTT and $\mathrm{TT}^{\mathrm{obs}}$, the logical relation is indexed by a level ℓ, which reflects the predicative nature of the universe hierarchy: reducibility is first defined at level 0 to characterize terms that inhabit the smallest universe $\mathcal{U}_0$, then this relation is used to define reducibility at level 1 for terms that live in $\mathcal{U}_1$ at most, and so on. Therefore, the whole definition is done by induction on ℓ. This construction seems deeply incompatible with an impredicative universe, as dependent products in Ω may mention arbitrarily large universes in their domain while still being at the bottom of the universe hierarchy. Thus replicating the usual definition for the impredicative universe leads to a non well-founded relation.

In this section, we show how to break out of that circularity in the case of an impredicative universe with definitional proof-irrelevance, by defining a weaker logical relation on impredicative types that does not require knowledge of the logical relation for the predicative hierarchy. However, a weaker logical relation means a weaker induction hypothesis, which makes the fundamental lemma harder to prove. We make up for this by having typing rules that have seemingly redundant hypotheses in the typing derivations. These hypotheses are used during the proof of the fundamental lemma, to recover information that cannot be collected in the logical relation.

In practice, the proof of the fundamental lemma requires our induction hypothesis to be stable under substitution. This leads us to further generalize reducibility to *validity*, which is defined as the closure of reducibility under substitution, and is noted with the $\Vdash^v_\ell$ turnstile (for instance, valid equality of terms is written $\Gamma \Vdash^v_\ell t \equiv u : A : s$). We can then finally show that all well-typed terms are valid by induction, which implies that they are reducible when specializing validity to the identity substitution, and thus normalizing. Finally, to get decidability of conversion for $\mathrm{CC}^{\mathrm{obs}}$, we introduce the notion of algorithmic conversion, noted $\Gamma \vdash t \cong u : A : s$, which is sound with respect to conversion and for which we can show that it is implied by the reducible equality. Fig. 7 summarizes the connections between those relations.

3.1 The Logical Relation for the Predicative Hierarchy

We start by presenting the logical relation for the relevant fragment, closely following Abel et al. [2018]; Pujet and Tabareau [2022]. For pedagogical reasons, we will present a somewhat informal version of the logical relation where some arguments are left implicit, and refer the interested reader to the Agda formalization. While our formal proof and the definition in *[LogicalRelation.agda]* only feature three levels, it should be quite clear that the same technique works for any finite number of levels.

The logical relation is defined by induction-recursion, as detailed in Fig. 8. The logical relation on types is defined inductively with seven constructors, one for every base type of $\mathrm{CC}^{\mathrm{obs}}$. Each

$$
\begin{array}{lcl}
\_ \Vdash_\ell \_ : \_ & : & (\Gamma : \mathsf{Context}) \to (t : \mathsf{Term}) \to (A : \mathsf{Term}) \to (s : \mathsf{Sort}) \to \mathsf{Set} \\
\_ \Vdash_\ell \_ : \_ & ::= & \Vdash_{\mathsf{ne}} : \forall\, \Gamma\ t\ s \to (\Gamma \Vdash_{\mathsf{ne},\ell} t : s) \to \Gamma \Vdash_\ell t : s \\
& | & \Vdash_{\mathsf{U}} : \forall\, \Gamma\ t\ i \to (\Gamma \Vdash_{\mathsf{U},\ell} t : \mathcal{U}_i) \to \Gamma \Vdash_\ell t : \mathcal{U}_i \\
& | & \Vdash_{\mathbb{N}} : \forall\, \Gamma\ t \to (\Gamma \Vdash_{\mathbb{N},\ell} t) \to \Gamma \Vdash_\ell t : \mathcal{U}_0 \\
& | & \Vdash_{\Pi} : \forall\, \Gamma\ t\ i \to (\Gamma \Vdash_{\Pi,\ell} t : \mathcal{U}_i) \to \Gamma \Vdash_\ell t : \mathcal{U}_i \\
& | & \Vdash_{\Omega} : \forall\, \Gamma\ t\ i \to (\Gamma \Vdash_{\Omega,\ell} t : \mathcal{U}_i) \to \Gamma \Vdash_\ell t : \mathcal{U}_i \\
& | & \Vdash_{\forall} : \forall\, \Gamma\ t \to (\Gamma \Vdash_{\forall,\ell} t) \to \Gamma \Vdash_\ell t : \Omega \\
& | & \Vdash_{\&} : \forall\, \Gamma\ t \to (\Gamma \Vdash_{\&,\ell} t) \to \Gamma \Vdash_\ell t : \Omega \\
& | & \Vdash_{\bot} : \forall\, \Gamma\ t \to (\Gamma \Vdash_{\bot,\ell} t) \to \Gamma \Vdash_\ell t : \Omega
\end{array}
$$

$$
\begin{array}{lcl}
\_ \Vdash_\ell \_ \equiv \_ : \_ & : & (\Gamma : \mathsf{Context}) \to (A : \mathsf{Term}) \to (B : \mathsf{Term}) \to (s : \mathsf{Sort}) \to \{\Gamma \Vdash_\ell A : s\} \to \mathsf{Set} \\
\Gamma \Vdash_\ell A \equiv B : s\ \{\Vdash_X\} & = & \Gamma \Vdash_{X,\ell} A \equiv B : s \qquad \text{where } X \in \{\mathsf{ne}, \mathsf{U}, \mathbb{N}, \Pi, \Omega, \forall, \&, \bot\}
\end{array}
$$

$$
\begin{array}{lcl}
\_ \Vdash_\ell \_ : \_ : \_ & : & (\Gamma : \mathsf{Context}) \to (t : \mathsf{Term}) \to (A : \mathsf{Term}) \to (s : \mathsf{Sort}) \to \{\Gamma \Vdash_\ell A : s\} \to \mathsf{Set} \\
\Gamma \Vdash_\ell t : A : s\ \{\Vdash_X\} & = & \Gamma \Vdash_{X,\ell} t : A : s \qquad \text{where } X \in \{\mathsf{ne}, \mathsf{U}, \mathbb{N}, \Pi, \Omega, \forall, \&, \bot\}
\end{array}
$$

$$
\begin{array}{lcl}
\_ \Vdash_\ell \_ \equiv \_ : \_ : \_ & : & (\Gamma : \mathsf{Context}) \to (t\ u : \mathsf{Term}) \to (A : \mathsf{Term}) \to (s : \mathsf{Sort}) \to \{\Gamma \Vdash_\ell A : s\} \to \mathsf{Set} \\
\Gamma \Vdash_\ell t \equiv u : A : s\ \{\Vdash_X\} & = & \Gamma \Vdash_{X,\ell} t \equiv u : A : s \qquad \text{where } X \in \{\mathsf{ne}, \mathsf{U}, \mathbb{N}, \Pi, \Omega, \forall, \&, \bot\}
\end{array}
$$

Fig. 8. Inductive-recursive presentation of the logical relation

constructor involves an auxiliary predicate of the form $\Gamma \Vdash_{X,\ell} t$ that packs the appropriate information for the corresponding type. The logical relations on terms and equalities are simultaneously defined by recursion on $\Gamma \Vdash_\ell A : s$, using more auxiliary predicates. This technically means that the logical relation on terms and equalities should have an extra argument of type $\Gamma \Vdash_\ell A : s$, but since it will turn out that all proofs of $\Gamma \Vdash_\ell A : s$ give rise to equivalent judgments (cf *[Irrelevance.agda]*), we freely omit this argument.

We now give the definition of all these auxiliary predicates, starting with the predicates for neutral types.

Neutral Types.

$$\frac{\Gamma \vdash A \Rightarrow^* N : s \qquad \mathrm{neutral}\, N}{\Gamma \Vdash_{\mathsf{ne},\ell} A : s}$$

This rule states that A is reducible to a neutral type: the notation $\Gamma \vdash A \Rightarrow^* N : s$ means that A reduces to N in a finite number of steps (possibly zero), and that both A and N are well-formed types of sort s in context Γ. When A is reducible to a neutral type, we define:

- $\Gamma \Vdash_{\mathsf{ne}} A \equiv B : s$ if there is a neutral term M such that $\Gamma \vdash B \Rightarrow^* M : s$ and $\Gamma \vdash N \equiv M : s$,
- $\Gamma \Vdash_{\mathsf{ne}} t : A : s$ if there is a neutral term n such that $\Gamma \vdash t \Rightarrow^* n : N$,
- $\Gamma \Vdash_{\mathsf{ne}} t \equiv u : A : s$ if there are neutral terms n, m such that $\Gamma \vdash t \Rightarrow^* n : N$ and $\Gamma \vdash u \Rightarrow^* m : N$, and $\Gamma \vdash n \equiv m : N$.

Note that the definition of reducible neutral types and terms does not recursively call the logical relation, thus it can be defined outside of the inductive-recursive definition.

Natural Numbers.

$$\frac{\Gamma \vdash A \Rightarrow^* \mathbb{N} : \mathcal{U}_0}{\Gamma \Vdash_{\mathbb{N},\ell} A}$$

When A is reducible to $\mathbb{N}$, we define:

- $\Gamma \Vdash_{\mathbb{N}} A \equiv B$ if $\Gamma \vdash B \Rightarrow^* \mathbb{N} : \mathcal{U}_0$,

- $\Gamma \Vdash_{\mathbb{N}} t : A$ if there is a weak-head normal form t' such that $\Gamma \vdash t \Rightarrow^* t' : \mathbb{N}$ and $\Gamma \Vdash_{\mathbb{N}t} t'$, which is inductively defined by

$$\frac{}{\Gamma \Vdash_{\mathbb{N}t} 0} \qquad \frac{\Gamma \Vdash_{\mathbb{N}t} t}{\Gamma \Vdash_{\mathbb{N}t} \mathsf{S}\, t} \qquad \frac{\Gamma \vdash n : \mathbb{N} \qquad n \text{ is neutral}}{\Gamma \Vdash_{\mathbb{N}t} n}$$

- $\Gamma \Vdash_{\mathbb{N}} t \equiv u : A$ if there are weak-head normal forms t', u' such that $\Gamma \vdash t \Rightarrow^* t' : \mathbb{N}$ and $\Gamma \vdash u \Rightarrow^* u' : \mathbb{N}$, and $\Gamma \Vdash_{\mathbb{N}t=} t' \equiv u'$, which is inductively defined by

$$\frac{}{\Gamma \Vdash_{\mathbb{N}t=} 0 \equiv 0} \qquad \frac{\Gamma \Vdash_{\mathbb{N}t=} t \equiv u}{\Gamma \Vdash_{\mathbb{N}t=} \mathsf{S}\, t \equiv \mathsf{S}\, u} \qquad \frac{\Gamma \vdash n \equiv m : \mathbb{N} \qquad n, m \text{ are neutral}}{\Gamma \Vdash_{\mathbb{N}t=} n \equiv m}$$

Again, those definitions do not mention the logical relation and can be defined *a priori*.

Predicative Universes.

$$\frac{\Gamma \vdash A \Rightarrow^* \mathcal{U}_i : \mathcal{U}_{i+1}}{\Gamma \Vdash_{\mathsf{U},\ell} A : \mathcal{U}_{i+1}}\, i<\ell$$

When A is reducible to a predicative universe, we define:

- $\Gamma \Vdash_{\mathsf{U},\ell} A \equiv B : \mathcal{U}_{i+1}$ if $\Gamma \vdash B \Rightarrow^* \mathcal{U}_i : \mathcal{U}_{i+1}$,
- $\Gamma \Vdash_{\mathsf{U},\ell} t : A : \mathcal{U}_{i+1}$ if there is a weak-head normal form t' such that $\Gamma \vdash t \Rightarrow^* t' : \mathcal{U}_i$, and $\Gamma \Vdash_i t : \mathcal{U}_i$
  (which is already defined by induction hypothesis, since $i < \ell$).
- $\Gamma \Vdash_{\mathsf{U},\ell} t \equiv u : A : \mathcal{U}_{i+1}$ if there are weak-head normal forms t', u' such that $\Gamma \vdash t \Rightarrow^* t' : \mathcal{U}_i$ and $\Gamma \vdash u \Rightarrow^* u' : \mathcal{U}_i$, and $\Gamma \Vdash_i t : \mathcal{U}_i$, $\Gamma \Vdash_i u : \mathcal{U}_i$, and $\Gamma \Vdash_i t \equiv u : \mathcal{U}_i$.

Proof-Relevant Dependent Function Types.

$$\frac{\begin{array}{c}\Gamma \vdash A \Rightarrow^* \Pi^{s,\mathcal{U}_i}(x : F).\, G : \mathcal{U}_j \qquad \mathcal{R}(s,\mathcal{U}_i) = \mathcal{U}_j \qquad \Gamma \vdash F : s \\ \Gamma, x : F \vdash G : \mathcal{U}_i \qquad \forall(\rho : \Delta \sqsubseteq \Gamma),\ \Delta \Vdash_\ell F[\rho] : s \qquad \forall(\rho : \Delta \sqsubseteq \Gamma),\ \Delta \Vdash_\ell a : F[\rho] : s \ \to\ \Delta \Vdash_\ell G[\rho, a] : \mathcal{U}_i \\ \forall(\rho : \Delta \sqsubseteq \Gamma),\ \Delta \Vdash_\ell a : F[\rho] : s \ \to\ \Delta \Vdash_\ell b : F[\rho] : s \ \to\ \Delta \Vdash_\ell a \equiv b : F[\rho] : s \ \to\ \Delta \Vdash_\ell G[\rho, a] \equiv G[\rho, b] : \mathcal{U}_i\end{array}}{\Gamma \Vdash_{\Pi,\ell} A : \mathcal{U}_j}$$

This rule states that A is reducible to a dependent function type. We introduced quite a bit of notation here: the notation $\to$ stands for the implication in the meta-theory, *i.e.*, the theory of Agda. We note $\forall(\rho : \Delta \sqsubseteq \Gamma)$ for the meta-theoretical quantification on an arbitrary well-formed context Δ and a weakening ρ that turns Δ into Γ. Applying such a weakening on the free variables of a term F that is a well-typed in context Γ results in a term $F[\rho]$ that is well-typed in context Δ. Given a term G in the extended context $\Gamma, x : F$ and a term $\Delta \vdash a : F[\rho]$, we can apply ρ on the free variables of G except for x to get a term in $\Delta, x : F[\rho]$, and then substitute x with a to get a term in Δ. The result is noted $G[\rho, a]$.

With those notations, a term A is reducible to a dependent function type when (i) it reduces to $\Pi(x : F).\, G$, (ii) F is a reducible type, (iii) given any reducible term a at type F, $G[a]$ is a reducible type and (iv) when a and b are reducibly equal at type B, $G[a]$ and $G[b]$ are reducibly equal types. Moreover, all these reducibility premises should hold under any weakening ρ. Note that the definition above defines reducibility of a type using reducibility of terms, which explains the need for an inductive-recursive definition.

When A is reducible to a dependent function type, we define:

- $\Gamma \Vdash_{\Pi,\ell} A \equiv B : \mathcal{U}_j$ if there are terms F' and G' such that
  - $\Gamma \vdash B \Rightarrow^* \Pi(x : F').\, G' : \mathcal{U}_j$ and $\Gamma \vdash \Pi(x : F).\, G \equiv \Pi(x : F').\, G' : \mathcal{U}_j$
  - $\forall(\rho : \Delta \sqsubseteq \Gamma),\ \Delta \Vdash_\ell F[\rho] \equiv F'[\rho] : s$
  - $\forall(\rho : \Delta \sqsubseteq \Gamma),\ \Delta \Vdash_\ell a : F[\rho] : s \ \to\ \Delta \Vdash_\ell G[\rho, a] \equiv G'[\rho, a] : \mathcal{U}_i$.
- $\Gamma \Vdash_{\Pi,\ell} t : A : \mathcal{U}_j$ if there is a weak-head normal form t' such that

  - $\Gamma \vdash t \Rightarrow^* t' : \Pi(x : F).\, G$
  - $\forall(\rho : \Delta \sqsubseteq \Gamma),\ \ \Delta \Vdash_\ell a : F[\rho] : s \ \rightarrow\ \Delta \Vdash_\ell t'[\rho]\ a : G[\rho, a] : \mathcal{U}_i$
  - $\forall(\rho : \Delta \sqsubseteq \Gamma),\ \ \Delta \Vdash_\ell a : F[\rho] : s \ \rightarrow\ \Delta \Vdash_\ell b : F[\rho] : s \ \rightarrow\ \Delta \Vdash_\ell a \equiv b : F[\rho] : s$
    $\rightarrow\ \Delta \Vdash_\ell t'[\rho]\ a \equiv t'[\rho]\ b : G[\rho, a] : \mathcal{U}_i.$
- $\Gamma \Vdash_{\Pi,\ell} t \equiv u : A : \mathcal{U}_j$ if there are weak-head normal forms t', u' such that
  - $\Gamma \vdash t \Rightarrow^* t' : \Pi(x : F).\, G$ and $\Gamma \vdash u \Rightarrow^* u' : \Pi(x : F).\, G$
  - $\Gamma \vdash t' \equiv u' : \Pi(x : F).\, G$
  - $\Gamma \Vdash_\ell t : A : \mathcal{U}_j$ and $\Gamma \Vdash_\ell u : A : \mathcal{U}_j$
  - $\forall(\rho : \Delta \sqsubseteq \Gamma),\ \ \Delta \Vdash_\ell a : F[\rho] : s \ \rightarrow\ \Delta \Vdash_\ell t'[\rho]\ a \equiv u'[\rho]\ a : G[\rho, a] : \mathcal{U}_i.$

The three definitions above make sense because we know that when A is reducible to a dependent function type, we also know that its domain F and codomain G are reducible.

3.2 The Logical Relation for the Impredicative Universe

We now turn to the definition of the relation for the proof-irrelevant fragment, which must be defined independently from the logical relation to avoid a non well-founded definition.

Since there is no computation whatsoever in proof-irrelevant types, their inhabitants don't interact with other terms. This allows us to give a generic definition for reducibility of terms and term equality, that will work for any $X \in \{\bot, \forall, \&\}$:

- $\Gamma \Vdash_{X,\ell} t : A : \Omega$ when $\Gamma \vdash t : A$.
- $\Gamma \Vdash_{X,\ell} t \equiv u : A : \Omega$ when $\Gamma \vdash t : A$ and $\Gamma \vdash u : A$.

This means that if A is in Ω, then the logical relation does not need to collect any information on inhabitants of A, save for the fact that they are well-typed. And this applies to equality too, since any two inhabitants of A are always convertible.

We still need to define the reducibility of types and type equality for $\bot$, impredicative dependent products and dependent conjunctions. We do not treat the case of $\top$, since it can be encoded as $\bot \rightarrow \bot$. Additionally, note that the observational equality is not featured in the the logical relation. This is because it does not play the role of a type constructor, but rather that of a destructor that computes by case analysis on types. Thus the only possibility for an observational equality to be a type in whnf is when it is a neutral type, a case that is already handled in our logical relation.

Empty type. The case of the empty type is easy, as it does not recursively call the logical relation.

$$\frac{\Gamma \vdash A \Rightarrow^* \bot : \Omega}{\Gamma \Vdash_{\bot,\ell} A}$$

When A is reducible to the empty type, we define $\Gamma \Vdash_{\bot,\ell} A \equiv B$ as $\Gamma \vdash B \Rightarrow^* \bot : \Omega$.

Impredicative Dependent Function Types. For the impredicative dependent function types, the situation is more complex, as we can not reproduce the definition of their predicative counterpart: it involves a recursive call to the logical relation for the domain type, which might live in a higher universe than the dependent function type. Consequently, we go for minimalism and only collect the fact that the domain and the codomain are well-typed.

$$\frac{\Gamma \vdash A \Rightarrow^* \Pi^{s,\Omega}(x : F).\, G : \Omega \qquad \Gamma \vdash F : s \qquad \Gamma, x : F \vdash G : \Omega}{\Gamma \Vdash_{\forall,\ell} A}$$

Similarly, when A is reducible to an impredicative dependent function type, we define reducible equality of A and B as the fact that B reduces to a convertible impredicative dependent function

type:

$$\frac{\Gamma \vdash B \Rightarrow^* \Pi(x : F').\, G' : \Omega \qquad \Gamma \vdash \Pi(x : F).\, G \equiv \Pi(x : F').\, G' : \Omega}{\Gamma \Vdash_{\forall,\ell} A \equiv B}$$

These definitions do not recursively call the logical relation, but as a result the collected invariants are much weaker than those for relevant dependent function types. We will see in Section 3.3 how to circumvent this problem by using apparently redundant hypothesis of the type system.

Dependent conjunctions. The definition of the logical relation for dependent conjunctions is similar to the one for impredicative dependent function types. We define

$$\frac{\Gamma \vdash A \Rightarrow^* (x : F) \,\&\, G : \Omega \qquad \Gamma \vdash F : \Omega \qquad \Gamma, x : F \vdash G : \Omega}{\Gamma \Vdash_{\&,\ell} A}$$

$$\frac{\Gamma \vdash B \Rightarrow^* (x : F') \,\&\, G' : \Omega \qquad \Gamma \vdash (x : F) \,\&\, G \equiv (x : F') \,\&\, G' : \Omega}{\Gamma \Vdash_{\&,\ell} A \equiv B}$$

The only remaining case of our definition is the impredicative universe Ω.

The Impredicative Universe.

$$\frac{\Gamma \vdash A \Rightarrow^* \Omega : \mathcal{U}_0}{\Gamma \Vdash_{\Omega,\ell} A : \mathcal{U}_0}$$

When A is reducible to the impredicative universe, we define:

- $\Gamma \Vdash_{\Omega,\ell} A \equiv B : \mathcal{U}_i$ if $\Gamma \vdash B \Rightarrow^* \Omega : \mathcal{U}_i$.
- $\Gamma \Vdash_{\Omega,\ell} t : A : \mathcal{U}_0$ if there is a weak-head normal form t' such that $\Gamma \vdash t \Rightarrow^* t' : \Omega$, and $\Gamma \Vdash_\ell t : \Omega$
  (which is already defined as reducibility of impredicative types is defined beforehand).
- $\Gamma \Vdash_{\Omega,\ell} t \equiv u : A : \mathcal{U}_0$ if there are weak-head normal forms t', u' such that $\Gamma \vdash t \Rightarrow^* t' : \Omega$ and $\Gamma \vdash u \Rightarrow^* u' : \Omega$, and $\Gamma \Vdash_i t : \Omega$, $\Gamma \Vdash_i u : \Omega$, and $\Gamma \Vdash_i t \equiv u : \Omega$.

3.3 The Fundamental Lemma

The fundamental lemma expresses the completeness of the logical relation with respect to typing *[Fundamental.agda]*. But before starting our induction, we need to generalize our hypothesis—the logical relation—yet a bit more: we will prove reducibility under any well-formed substitution. Following [Abel et al. 2018], we use a new instance of induction-recursion to define the notion of *validity* $\Vdash^v$, which will serve as our actual induction hypothesis *[Substitution.agda]*. A type A is said to be valid in context Γ if it satisfies the following rule:

$$\frac{\forall(\sigma : \Delta \to \Gamma),\, \Vdash^v \Delta \;\to\; \Delta \Vdash^s \sigma : \Gamma \;\to\; \begin{cases} \Delta \Vdash_\ell A[\sigma] : s \\ \forall \sigma'.\, \Delta \Vdash^s \sigma' : \Gamma \;\to\; \Delta \Vdash^s \sigma \equiv \sigma' : \Gamma \;\to\; \Delta \Vdash_\ell A[\sigma] \equiv A[\sigma'] : s \end{cases}}{\Gamma \Vdash^v_\ell A : s}$$

In this definition, the notation $\forall(\sigma : \Delta \to \Gamma)$ means that we quantify over an arbitrary substitution σ from Δ to Γ. The context Δ is assumed to be a *valid context* (noted $\Vdash^v \Delta$, meaning that the context is composed of valid types) and the substitution σ is assumed to be a *valid substitution* (noted $\Delta \Vdash^s \sigma : \Gamma$, meaning that it substitutes only reducible terms). Similarly, two valid substitutions σ and σ' are *validly equal*, noted $\Delta \Vdash^s \sigma \equiv \sigma' : \Gamma$, when they substitute reducibly equal terms.

There are similar notions of validity for terms, equality of types and equality of terms. For instance, when A is a valid type, the validity of a term t at A is defined as

$$\frac{\forall(\sigma : \Delta \to \Gamma),\ \Vdash^v \Delta \to \Delta \Vdash^s \sigma : \Gamma \to \begin{cases} \Delta \Vdash_\ell t[\sigma] : A[\sigma] : s \\ \forall \sigma'.\ \Delta \Vdash^s \sigma' : \Gamma \to \Delta \Vdash^s \sigma \equiv \sigma' : \Gamma \to \Delta \Vdash_\ell t[\sigma] \equiv t[\sigma'] : A[\sigma] : s \end{cases}}{\Gamma \Vdash^v_\ell t : A : s}$$

The lemma is proven using a mutual induction on derivations for well-typed types, terms, equality of types and equality of terms. We only present the statement on well-typed terms as we only explain the difference from the proof of Abel et al. [2018]; Pujet and Tabareau [2022] on the impredicative fragment. Note that because validity of a term can only be defined if the context and the type are themselves valid, the fundamental lemma actually provides validity of every component of the typing judgment.

Lemma 3.1 (Fundamental lemma (for terms) *[Fundamental.agda]*). *If $\Gamma \vdash t : A : s$, then there is a level ℓ such that $\Vdash^v \Gamma$, $\Gamma \Vdash^v_\ell A : s$ and $\Gamma \Vdash^v_\ell t : A : s$.*

Proof. The theorem is proven by induction on the derivation. The only cases where the proof differs from Abel et al. [2018]; Pujet and Tabareau [2022] is when a rule builds an inhabitant of a proof-irrelevant type. In this case, we need to show that (i) the proof-irrelevant type is valid, and (ii) that the term is well-typed. Showing that the term is well-typed is always straightforward, but the validity of its type can be more difficult to obtain.

In the case of the application of functions (Rule App), the usual predicative way to get the fact that $B[x := u]$ is valid is to extract it from the information collected by the validity of the dependent function type (which is known by induction hypothesis). In the impredicative case however, we do not get such expansive information from the validity of the dependent function type, because validity of a type in the impredicative universe cannot recursively mention validity of the codomain. This is precisely where the apparently redundant assumptions come into play: from the induction hypothesis on the additional assumptions, we also know that B is valid as a type family over A. Since validity is defined as reducibility under any valid substitution, we can get the reducibility of $B[x := u]$.

□

A direct consequence of the fundamental lemma is that any well-typed term has a weak-head normal form. Another consequence is that any closed term of type $\mathbb{N}$ reduces to a normal form of type $\mathbb{N}$. Thus, to obtain canonicity for the integers of $\mathrm{CC}^{\mathrm{obs}}$, we just need to know that there is no neutral term of type $\mathbb{N}$ in an empty context *[Canonicity.agda]*.

Unfortunately, as in [Pujet and Tabareau 2022], the logical relation is not sufficient to establish this. In fact, as we will see in Section 4, there is a good reason for that: the consistency and canonicity theorems for $\mathrm{CC}^{\mathrm{obs}}$ have a remarkably high proof-theoretical strength due to impredicativity, whereas the logical relation argument can be developed in a predicative meta-theory. In Section 5, we will supplement the normalization result with a model of $\mathrm{CC}^{\mathrm{obs}}$ in an impredicative meta-theory, from which we will deduce the consistency and canonicity theorems.

3.4 Decidability of Conversion

Besides proving weak-head normalization of well-typed terms, the main point of the fundamental lemma is to be able to show that conversion is decidable in $\mathrm{CC}^{\mathrm{obs}}$. To do so, we define an algorithmic relation $\Gamma \vdash t \cong u : A : s$ which is supposed to simulate convertibility of two terms t and u, while being easier to decide *[Conversion.agda]*. This algorithmic conversion keeps track of the relevance information, and uses it to either give an immediate answer in case t and u are irrelevant, or compute their weak-head normal forms, then compare their head constructor, and possibly apply

the algorithm recursively in case t and u are relevant. Correctness of this algorithmic conversion is direct as the rules used are subsumed by the general conversion judgement *[Soundness.agda]*.

Showing that the algorithmic conversion is also complete is more complex. It basically amounts to replaying the fundamental lemma with a definition of the logical relation that uses algorithmic conversion instead of the general conversion *[Completeness.agda]*. In our formal proof, we follow Abel et al. [2018] in factoring the two instances of the fundamental lemma by defininig a generic interface for both algorithmic conversion and typed conversion, and using this interface in the definition of the logical relation *[EqualityRelation.agda]*.

Theorem 3.2 (Equivalence of conversion and algorithmic conversion). *Given two terms t and u such that $\Gamma \vdash t : A : s$ and $\Gamma \vdash u : A : s$, we have that*

$$\Gamma \vdash t \equiv u : A : s \iff \Gamma \vdash t \cong u : A : s.$$

It still remains to provide a decision procedure for the algorithmic conversion *[Decidable.agda]*. Given two terms t and u well-typed at $A : s$ in context Γ, we can apply the fundamental lemma plus the reflexivity of the logical relation to get the fact that $\Gamma \Vdash_\ell t \equiv t : A : s$ and similarly for u. Because reducibly equal terms are also algorithmically convertible, we have $\Gamma \vdash t \cong t : A : s$ (and similarly for u). Then the decision procedure is done by double induction on the proofs that t and u are reflexive for the algorithmic conversion. The idea is that t and u are convertible if and only if the two reflexive proofs are the same.

Note that the proof that t is algorithmically equal to itself contains the fact that t strongly normalizes, because t can be recursively put in whnf.

3.5 Removing Redundant Premises

So far, we established the fundamental lemma on a type system with many redundant premises (see for instance the rule App). In practice, we often want to work with a more economic presentation that for instance may lead to more efficient type-checking algorithms. Although these redundant premises played an essential role in our proof, once the fundamental lemma is established we can show that they can safely be removed.

First, we establish the typing validity (a.k.a. typing regularity) by combining the fundamental lemma with the fact that reducible types are well-formed and reducible terms are well-typed:

Corollary 3.3 (Typing Validity *[Syntactic.agda]*). *If $\Gamma \vdash t : A : s$ then $\Gamma \vdash A : s$ and similarly, if $\Gamma \vdash t \equiv u : A : s$ then $\Gamma \vdash A : s$, $\Gamma \vdash t : A : s$ and $\Gamma \vdash u : A : s$.*

And now that we are equipped with this corollary, as well as some inversion lemmas that we obtained from the logical relation, we know enough to design a more economic presentation of $\mathrm{CC}^{\mathrm{obs}}$. The economic type system and the proof of equivalence are spelled out in full detail in *[NonParanoidTyping.agda]*. For instance, the rule for application may be reduced to

$$\textsc{App-eco}\quad \frac{\Gamma \vdash t : \Pi(x : A).\, B : \mathcal{R}(s,s') \qquad \Gamma \vdash u : A : s}{\Gamma \vdash t\ u : B[x := u] : s'}$$

4 ANALYSIS OF THE NORMALIZATION PROOF

The informal proof presented in Section 3, and formalized in Agda, outlines the construction of a normalization model in a rather powerful metatheory: Martin-Löf Type Theory extended with induction-recursion. However, in order to control the computational complexity of the proof terms more finely, we can try to weaken this metatheory as much as possible.

4.1 The Computational Power of $\mathrm{CC}^{\mathrm{obs}}$

To establish a simple lower bound, we can start by noting that Martin-Löf Type Theory is a subset of $\mathrm{CC}^{\mathrm{obs}}$, if we include the same amount of inductive types in both systems. Indeed, dependent products and the universes are handled in the exact same way, and Pujet and Tabareau [2022] explain how to add basic inductive types to $\mathrm{CC}^{\mathrm{obs}}$ such as dependent sums, the inductive equality or W-types. Therefore, a normalization proof for $\mathrm{CC}^{\mathrm{obs}}$ can also act as a normalization proof for MLTT. And since normalization entails consistency for MLTT, it follows that our meta-theory needs to be at least as strong as MLTT—in fact, it turns out to be exactly what we need.

Theorem 4.1. *MLTT with $n + 4$ universes and a general scheme of indexed inductive types can prove normalization of $\mathrm{CC}^{\mathrm{obs}}$ with n universes and basic inductive types (Σ-types, W-types, identity).*

We defer the proof to Section 4.2. This theorem suggests that the normalization theorem for $\mathrm{CC}^{\mathrm{obs}}$ with the full hierarchy of universes is equivalent to normalization for MLTT over a weak fragment of arithmetic, but a complete proof of this fact would need a strengthened version of Theorem 4.1 that does not require general indexed inductive types. Remark that we cannot hope for a similar result for consistency or canonicity: while consistency and canonicity for MLTT follow from normalization, proving that $\mathrm{CC}^{\mathrm{obs}}$ is consistent requires the increased power of impredicativity.

Theorem 4.1 also provides us with some control over the integer functions of $\mathrm{CC}^{\mathrm{obs}}$:

Corollary 4.2. *Any integer function that can be expressed as a closed term of type $\mathbb{N} \to \mathbb{N}$ in $\mathrm{CC}^{\mathrm{obs}}$ can also be defined in MLTT with a general scheme for indexed inductive types.*

Proof. Note that any closed term f of type $\mathbb{N} \to \mathbb{N}$ in $\mathrm{CC}^{\mathrm{obs}}$ only mentions a finite number of universes. Thus, Theorem 4.1 provides us with a normalization function for the corresponding fragment of $\mathrm{CC}^{\mathrm{obs}}$, that computes a normal form when applied to $f\,n$ where n is a closed syntactic integer. If we compose it with a function that sends normal forms to the adequate integer, we obtain an integer function in MLTT, and canonicity ensures that it represents the same function as the original $\mathrm{CC}^{\mathrm{obs}}$ term. □

Corollary 4.2 might come off as a surprise, since $\mathrm{CC}^{\mathrm{obs}}$ is equipped with an impredicative universe, and impredicativity generally adds a great deal of proof-theoretical strength! And $\mathrm{CC}^{\mathrm{obs}}$ does possess this increased logical power, in fact. Indeed, it can represent many more functions as Ω-valued functional relations than MLTT. But this corollary shows that there is no way to extract them in the proof-relevant fragment, even though we have access to elimination principles for false propositions and equalities (using casts). Thus, the logical power of impredicativity is locked inside of Ω. This is in stark contrast with the Calculus of Inductive Constructions, in which it is possible to define closed terms of type $\mathbb{N} \to \mathbb{N}$ that leverage the power of impredicativity, using a principle called large elimination of singleton inductive types[2]. We discuss this principle with more detail in Section 6.

4.2 Fitting the Normalization Proof in MLTT

In this subsection, we give a proof of Theorem 4.1, by encoding the logical relation sketched in Section 3 without induction-recursion. This folklore technique has already been described in various informal settings [Abel et al. 2007; Abel and Coquand 2005; Sterling et al. 2019], but we go a step further and provide a formalization of the argument in Agda.[3] For the remainder of this section, we work in MLTT with indexed inductive types, but no induction-recursion, and we define

[2]The standard technique is to use the accessibility predicate as defined in https://coq.inria.fr/library/Coq.Init.Wf.html.

[3]The formalization can be found in https://github.com/loic-p/logrel-mltt/tree/without-IR. It is done for MLTT and not full $\mathrm{CC}^{\mathrm{obs}}$, but it does not make a difference as the crux is in the definition of the relation without induction-recursion.

a deep embedding of $\mathrm{CC}^{\mathrm{obs}}$. To avoid confusion between the meta-theory and the object theory, we will use Agda-style notations for the meta-theory.

From a bird's eye perspective, the normalization proof builds a model of $\mathrm{CC}^{\mathrm{obs}}$ in which well-formed types are interpreted as proof-relevant predicates on untyped terms, and induction-recursion is used to construct a universe in the model: we define the inductive predicate of reducible types simultaneously with recursive functions that associates reducibility predicates to a reducible type. On closer inspection, we realize that the proof still works if the reducibility predicate of a universe lives one universe higher than the reducibility predicates of the types it contains. This allows us to use *small* induction-recursion, which can be replaced by functional relations that only require simple indexed inductive types [Hancock et al. 2013].

Fig. 9 showcases what the relation-based definition looks like. For all $\ell \leq n$, we define an inductive relation R^ℓ between a context, an untyped term t (the reducible type), its sort, a predicate $P_=$ of types that are convertible to t, a predicate P_t which is the reducibility predicate associated to t, and a binary relation $P_{t=}$ that encodes convertibility of terms in P_t. The logical relation $\Vdash_\ell$ is then defined in terms of R^ℓ. The inductive relation features eight constructors that are each defined in Section 3, including R_Π which recursively calls the logical relation.

Note that the logical relation is built in stages:

- We first define an inductive relation R^0 that has cases for dependent products and all base types, except for proof-relevant universes: it only accounts for inhabitants of $\mathcal{U}_0$.
- From R^0, we define a reducibility predicate $P_{\mathcal{U}_0}$ for $\mathcal{U}_0$: a term t is in $P_{\mathcal{U}_0}$ if there exist $P_=, P_t, P_{t=}$ such that $R^0(t, P_=, P_t, P_{t=})$. Now that we have a predicate for $\mathcal{U}_0$, we can define a relation R^1 that accounts for inhabitants of $\mathcal{U}_1$. Of course, since $P_{\mathcal{U}_0}$ lives in Set_1, R^1 will land in Set_2.
- We repeat this process n times to obtain a relation R^n that handles our n universes.

Each step of this process requires an additional meta-theoretical universe level. This is only natural, since we are proving normalization for a theory that subsumes MLTT_n, a property from which we can deduce the consistency of MLTT_n. Therefore, Gödel's incompleteness theorem should apply and guarantee that we need more universes in the meta-theory than in the fragment we consider—that is, if the general indexed inductive types do not add logical power compared to to the theory of Martin-Löf [1975]. Note that we may in principle use only one additional meta-theoretical universe level but because of the use of record types to pack definitions, we actually need two.

Constructing this universe of reducible types is only the first half of the normalization proof. To complete the construction, we also need to encode validity without induction-recursion in MLTT and prove the fundamental lemma. We do not develop this here as it uses a similar construction and we refer the interested reader to the formalization for the full proof. We just mention that the definition of validity requires quantification over reducible substitution which again requires two additional meta-theoretical universe levels. In the end, we can justify n universes using $n + 4$ universes in Agda, hence the statement of Theorem 4.1.

5 SEMANTICS OF $\mathrm{CC}^{\mathrm{obs}}$

In this section, we turn to the construction of the *standard* set-theoretic model of $\mathrm{CC}^{\mathrm{obs}}$ by adapting the construction of Gratzer [2022] to our setting. This model serves two purposes: to show consistency of our theory on the one hand, and to ensure that $\mathrm{CC}^{\mathrm{obs}}$ is a good internal language for set theory on the other hand.

5.1 Deriving Consistency from a Model

As we already mentioned, the normalization model that we built in Section 3 is not strong enough to obtain consistency and canonicity for $\mathrm{CC}^{\mathrm{obs}}$. Indeed, this model interprets the proof-irrelevant

$$
\begin{array}{lcl}
\mathsf{R}^\ell & : & \mathsf{Context} \to \mathsf{Term} \to (\mathsf{Term} \to \mathsf{Set}_{\ell+1}) \to (\mathsf{Term} \to \mathsf{Set}_{\ell+1}) \to (\mathsf{Term} \to \mathsf{Term} \to \mathsf{Set}_{\ell+1}) \to \\
& & \mathsf{Set}_{\ell+2} \\
\mathsf{R}^\ell & ::= & \mathsf{R}_{\mathsf{ne}} : \forall\, \Gamma\ t \to (\Gamma \Vdash_{\mathsf{ne}} t) \to \mathsf{R}^\ell\ \Gamma\ t\ (\Gamma \Vdash_{\mathsf{ne}} t \equiv \_)\ (\Gamma \Vdash_{\mathsf{ne}} \_ : t)\ (\Gamma \Vdash_{\mathsf{ne}} \_ \equiv \_ : t) \\
& | & \mathsf{R}_{\mathsf{U}} : \forall\, \Gamma\ t \to (\Gamma \Vdash_{\mathsf{U}} t) \to \mathsf{R}^\ell\ \Gamma\ t\ (\Gamma \Vdash_{\mathsf{U}} t \equiv \_)\ (\Gamma \Vdash_{\mathsf{U}} \_ : t)\ (\Gamma \Vdash_{\mathsf{U}} \_ \equiv \_ : t) \\
& | & \mathsf{R}_{\mathbb{N}} : \forall\, \Gamma\ t \to (\Gamma \Vdash_{\mathbb{N}} t) \to \mathsf{R}^\ell\ \Gamma\ t\ (\Gamma \Vdash_{\mathbb{N}} t \equiv \_)\ (\Gamma \Vdash_{\mathbb{N}} \_ : t)\ (\Gamma \Vdash_{\mathbb{N}} \_ \equiv \_ : t) \\
& | & \mathsf{R}_{\Pi} : \forall\, \Gamma\ t \to (\Gamma \Vdash_{\Pi} t) \to \mathsf{R}^\ell\ \Gamma\ t\ (\Gamma \Vdash_{\Pi} t \equiv \_)\ (\Gamma \Vdash_{\Pi} \_ : t)\ (\Gamma \Vdash_{\Pi} \_ \equiv \_ : t) \\
& | & \mathsf{R}_{\Omega} : \forall\, \Gamma\ t \to (\Gamma \Vdash_{\Omega} t) \to \mathsf{R}^\ell\ \Gamma\ t\ (\Gamma \Vdash_{\Omega} t \equiv \_)\ (\Gamma \Vdash_{\Omega} \_ : t)\ (\Gamma \Vdash_{\Omega} \_ \equiv \_ : t) \\
& | & \mathsf{R}_{\forall} : \forall\, \Gamma\ t \to (\Gamma \Vdash_{\forall} t) \to \mathsf{R}^\ell\ \Gamma\ t\ (\Gamma \Vdash_{\forall} t \equiv \_)\ (\Gamma \Vdash_{\forall} \_ : t)\ (\Gamma \Vdash_{\forall} \_ \equiv \_ : t) \\
& | & \mathsf{R}_{\exists} : \forall\, \Gamma\ t \to (\Gamma \Vdash_{\&} t) \to \mathsf{R}^\ell\ \Gamma\ t\ (\Gamma \Vdash_{\&} t \equiv \_)\ (\Gamma \Vdash_{\&} \_ : t)\ (\Gamma \Vdash_{\&} \_ \equiv \_ : t) \\
& | & \mathsf{R}_{\bot} : \forall\, \Gamma\ t \to (\Gamma \Vdash_{\bot} t) \to \mathsf{R}^\ell\ \Gamma\ t\ (\Gamma \Vdash_{\bot} t \equiv \_)\ (\Gamma \Vdash_{\bot} \_ : t)\ (\Gamma \Vdash_{\bot} \_ \equiv \_ : t)
\end{array}
$$

All the functions whose name contains $\Vdash_{\mathsf{ne}}$, $\Vdash_{\mathsf{U}}$, $\Vdash_{\mathbb{N}}$, $\Vdash_{\Pi}$, $\Vdash_{\forall}$, $\Vdash_{\&}$, $\Vdash_{\bot}$ are as defined in Section 3.

$$
\begin{array}{lcl}
\_ \Vdash_\ell \_ : \_ & : & (\Gamma : \mathsf{Context}) \to (t : \mathsf{Term}) \to (A : \mathsf{Term}) \to (s : \mathsf{Sort}) \to \mathsf{Set}_{\ell+1} \\
\Gamma \Vdash_\ell t : s & = & (P_= : \mathsf{Term} \to \mathsf{Set}_\ell) \times (P_t : \mathsf{Term} \to \mathsf{Set}_\ell) \\
& & \times\ (P_{t=} : \mathsf{Term} \to \mathsf{Term} \to \mathsf{Set}_\ell) \times (\mathsf{R}^\ell\ \Gamma\ t\ s\ P_=\ P_t\ P_{t=})
\end{array}
$$

$$
\begin{array}{lcl}
\_ \Vdash_\ell \_ \equiv \_ : \_ & : & (\Gamma : \mathsf{Context}) \to (A : \mathsf{Term}) \to (B : \mathsf{Term}) \to (s : \mathsf{Sort}) \to \{\Gamma \Vdash_\ell A : s\} \to \mathsf{Set}_\ell \\
\Gamma \Vdash_\ell A \equiv B : s & = & P_=\ B
\end{array}
$$

$$
\begin{array}{lcl}
\_ \Vdash_\ell \_ : \_ : \_ & : & (\Gamma : \mathsf{Context}) \to (t : \mathsf{Term}) \to (A : \mathsf{Term}) \to (s : \mathsf{Sort}) \to \{\Gamma \Vdash_\ell A : s\} \to \mathsf{Set}_\ell \\
\Gamma \Vdash_\ell t : A : s & = & P_t\ t
\end{array}
$$

$$
\begin{array}{lcl}
\_ \Vdash_\ell \_ \equiv \_ : \_ : \_ & : & (\Gamma : \mathsf{Context}) \to (t\ u : \mathsf{Term}) \to (A : \mathsf{Term}) \to (s : \mathsf{Sort}) \to \{\Gamma \Vdash_\ell A : s\} \to \mathsf{Set}_\ell \\
\Gamma \Vdash_\ell t \equiv u : A : s & = & P_{t=}\ t\ u
\end{array}
$$

Fig. 9. Inductive encoding of the logical relation

proposition $\bot$ as the set of syntactical terms with type $\bot$, which does not grant any kind of control over its proofs in the empty context.

This was already the case in the work of Pujet and Tabareau [2022] on $\mathsf{TT}^{\mathsf{obs}}$, so they supplemented their normalization proof with a model in the category of sets (presented as setoids), using induction-recursion to interpret universes as sets of codes *à la Tarski* defined mutually with eliminators and coercion functions. In that model, $\bot$ is actually interpreted as the empty set, which proves consistency of $\mathsf{TT}^{\mathsf{obs}}$: we can use it to show that there is no term of type $\bot$ in the empty context.

5.2 $\mathsf{CC}^{\mathsf{obs}}$ as an Internal Language for Sets

While the construction of Pujet and Tabareau [2022] does show consistency of $\mathsf{TT}^{\mathsf{obs}}$, it arguably falls short of presenting $\mathsf{TT}^{\mathsf{obs}}$ as an internal language for classical mathematics. Indeed, the inductive-recursive construction of the universe of codes is designed to only account for the codes that come from the syntax of $\mathsf{TT}^{\mathsf{obs}}$. In other words, the statement

$$\Pi(A : s)(x\ y\ z : A).\ x \sim_A y \to y \sim_A z \to x \sim_A z$$

only states transitivity of equality for sets that are built from the type formers of $\mathsf{TT}^{\mathsf{obs}}$, when interpreted in their model.

This state of affairs is obviously not ideal, as readers who are not willing to accept $\mathsf{CC}^{\mathsf{obs}}$ as a new foundation of mathematics would probably be more inclined to use it if they knew that any theorem they proved in $\mathsf{CC}^{\mathsf{obs}}$ is also true for classical set theory. Thus, instead of replicating this

construction in an impredicative meta-theory (to be able to interpret Ω), we seize the opportunity to solve two problems at once, and present a model in classical set theory that gives a much more sensible meaning to statements in $\mathrm{CC}^{\mathrm{obs}}$ while still being sufficient to show consistency.

5.3 Constructing the Set-theoretical Model

We work in ZF set theory with a countable hierarchy of Grothendieck universes $\mathbf{V}_0, \mathbf{V}_1, \ldots$ We call Ω the lattice of truth values (or in other words, the lattice of subsets of the singleton set $\{*\}$), with $\emptyset$ as its minimal value and given $p \in \Omega$ we write $\mathrm{val}\ p$ for the associated set $\{x \in \{*\} \mid p\}$. Even though we stay in the world of set theory for the duration of this section, we remain type theorists, and as such we use dependent products, dependent sums and inductive types in this section. They should be understood as their standard interpretation in the set-theoretic model of type theory. In an attempt to minimize confusion, we change our notations a little bit: we use $(a \in A) \to (B\ a)$ for the set-theoretic dependent product, $(a \in A) \times (B\ a)$ for the the set-theoretic dependent sum and $\boldsymbol{\omega}$ for the set-theoretic integers.

The central ingredient of our standard model is the interpratation of the proof-relevant universes of $\mathrm{CC}^{\mathrm{obs}}$. It is tempting to define them directly as Grothendieck universes, but unfortunately this does not work: in $\mathrm{CC}^{\mathrm{obs}}$, type constructors are injective with respect to the observational equality (for instance, $(A \to B) \sim (A' \to B')$ implies that $A \sim A'$ and $B \sim B'$) while set-theoretic function spaces collapse too much information for this to be possible ($\emptyset \to \{*\}$ and $\emptyset \to \boldsymbol{\omega}$ are identical).

This is not too difficult to fix, however: we can define a hierarchy of sets $\mathbf{U}_0, \mathbf{U}_1 \ldots$ that have the same element as the Grothendieck universes, but decorated with labels that keep track of how they were constructed. The most natural way to do this from a type theorist's perspective is to build an inductive predicate $\mathbf{U}_i^{\varepsilon}$ over $\mathbf{V}_i$ and then interpret the universes of $\mathrm{CC}^{\mathrm{obs}}$ as $\mathbf{U}_i = (X \in \mathbf{V}_i) \times (\mathbf{U}_i^{\varepsilon}\ X)$, as described in Fig. 10. Note how the constructor $\mathsf{c}_{\mathsf{emb}}$ builds a proof of $\mathbf{U}_i^{\varepsilon}\ X$ for any (small) set X, so that statements that quantify over $\mathcal{U}_i$ apply to all appropriately-sized sets of the model. This also implies that there might be several codes for the same set: for instance $(\mathbb{N} \to \mathbb{N}\ ;\ \mathsf{c}_{\mathsf{emb}})$ and $(\mathbb{N} \to \mathbb{N}\ ;\ \mathsf{c}_{\Pi\mathsf{U}\mathsf{U}}\ \ldots)$ are both codes for the set $\mathbb{N} \to \mathbb{N}$, but only the second one remembers that it has been built as a function space.

Remark that the set-theoretic equality between codes matches closely the equality of $\mathrm{CC}^{\mathrm{obs}}$: two codes can only be equal if they pack the same witness of $\mathbf{U}_i^{\varepsilon}$, which means that they have been built in the same way. Moreover, if we compare two codes of the form $(X \to Y;\ \mathsf{c}_{\Pi\mathsf{U}\mathsf{U}}\ \ldots)$, the equality of the second member means that the domains and the codomains of the function spaces are equal—we recover the injectivity of the type formers.

5.4 Interpreting the Syntax of $\mathrm{CC}^{\mathrm{obs}}$

Now that we know how to deal with the universes, it only remains to spell out the interpretation of the full syntax of $\mathrm{CC}^{\mathrm{obs}}$. In the standard fashion [Hofmann 1995], we define interpretation functions that send

- a context Γ to a set $[\![\Gamma]\!]$, and
- a pair of a term t and a context Γ to a set $[\![t]\!]_\Gamma$ indexed over $[\![\Gamma]\!]$.

Since these functions are defined on raw syntax without any guarantee of well-typedness, we cannot expect every term to have a sensible interpretation so we need to make them *partial* functions. We will be able to prove that all well-typed terms admit an interpretation *a posteriori* by induction on the derivations.

As usual, contexts are interpreted as telescopes: $[\![\bullet]\!] = \{*\}$ and $[\![\Gamma, x : A]\!] = (\rho : [\![\Gamma]\!]) \times ([\![A]\!]_\Gamma\ \rho)$ when $[\![A]\!]_\Gamma$ is defined. Given $\rho \in [\![\Gamma]\!]$ and a variable x that appears in Γ, we write $\rho(x)$ for the corresponding projection. The interpretation of the terms is defined in Fig. 11. The proof-relevant

$$
\begin{array}{lcl}
\mathbf{U}_i^\varepsilon & : & \mathbf{V}_i \to \mathbf{V}_{i+1} \\
\mathbf{U}_i^\varepsilon & ::= & \mathsf{c}_{\mathsf{emb}} : (X \in \mathbf{V}_i) \to \mathbf{U}_i^\varepsilon\ X \\
& | & \mathsf{c}_{\Pi\mathsf{U}\mathsf{U}} : \{j,k \in \boldsymbol{\omega} \mid \max(j,k) \le i\} \to (A \in \mathbf{V}_j) \to (A_\varepsilon \in \mathbf{U}_i^\varepsilon\ A) \\
& & \qquad \to (B \in (A \to \mathbf{V}_k)) \to (B_\varepsilon \in ((a \in A) \to \mathbf{U}_i^\varepsilon\ (B\ a))) \\
& & \qquad \to \mathbf{U}_i^\varepsilon\ ((a \in A) \to B\ a) \\
& | & \mathsf{c}_{\Pi\Omega\mathsf{U}} : (A \in \Omega) \\
& & \qquad \to (B \in (\mathsf{val}\ A \to \mathbf{V}_i)) \to (B_\varepsilon \in ((a \in \mathsf{val}\ A) \to \mathbf{U}_i^\varepsilon\ (B\ a))) \\
& & \qquad \to \mathbf{U}_i^\varepsilon\ ((a \in \mathsf{val}\ A) \to B\ a) \\
& | & \mathsf{c}_{\mathsf{U}} : \{j \in \boldsymbol{\omega} \mid j < i\} \to \mathbf{U}_i^\varepsilon\ \mathbf{U}_j \\
& | & \mathsf{c}_{\Omega} : \mathbf{U}_i^\varepsilon\ \Omega \\
\\
\mathbf{U}_i & := & (X \in \mathbf{V}_i) \times (\mathbf{U}_i^\varepsilon\ X) \\
\\
\mathsf{el} & : & \mathbf{U}_i \to \mathbf{V}_i \\
\mathsf{el}\ (X\ ;\ X_\varepsilon) & := & X
\end{array}
$$

Fig. 10. The universe of codes, defined with set-theoretic inductive types

fragment follows the standard interpretation of dependent type theory, and the observational equality is interpreted as the extensional equality of set-theory. Note how the irrelevant proofs are all interpreted as $*$, the inhabitant of the singleton set. This forces us to interpret cast as doing nothing, and $\bot\text{-elim}(A,t)$ is not even interpreted since it can only be formed in inconsistent contexts, which are empty in the model.

In order to prove the soundness of our interpretation, we need to extend it to weakenings and substitutions between contexts. Assume Γ and Δ are a syntactical contexts, and A and t are syntactical terms. In case $[\![\Gamma, x : A, \Delta]\!]$ and $[\![\Gamma, \Delta]\!]$ are well-defined, let π_A be the projection:

$$
\begin{array}{c}
\pi_A : [\![\Gamma, x : A, \Delta]\!] \to [\![\Gamma, \Delta]\!] \\
(\vec{x_\Gamma}, x_A, \vec{x_\Delta}) \mapsto (\vec{x_\Gamma}, \vec{x_\Delta}).
\end{array}
$$

In case $[\![\Gamma, \Delta[x := t]]\!]$ and $[\![\Gamma, x : A, \Delta]\!]$ are well-defined, we define the function σ_t by:

$$
\begin{array}{c}
\sigma_t : [\![\Gamma, \Delta[x := t]]\!] \to [\![\Gamma, x : A, \Delta]\!] \\
(\vec{x_\Gamma}, \vec{x_\Delta}) \mapsto (\vec{x_\Gamma}, [\![t]\!]_\Gamma\ \vec{x_\Gamma}, \vec{x_\Delta}).
\end{array}
$$

Lemma 5.1 (Weakening). *π_A is the semantic counterpart to the weakening of A: for all terms u, when both sides are well defined, we have:*

$$[\![u]\!]_{\Gamma,x:A,\Delta} = [\![u]\!]_{\Gamma,\Delta} \circ \pi_A$$

Lemma 5.2 (Substitution). *σ_t is the semantic counterpart to the substitution by t: for all terms u, when both sides are well defined, we have:*

$$[\![u[x := t]]\!]_{\Gamma,\Delta[x:=t]} = [\![u]\!]_{\Gamma,x:A,\Delta} \circ \sigma_t$$

Theorem 5.3 (Soundness of the Standard Model).

(1) *If $\vdash \Gamma$ then $[\![\Gamma]\!]$ is defined.*
(2) *If $\Gamma \vdash A : \Omega$ then $[\![A]\!]_\Gamma\ \rho$ is a proposition for all $\rho \in [\![\Gamma]\!]$.*
(3) *If $\Gamma \vdash A : \mathcal{U}_i$ then $[\![A]\!]_\Gamma\ \rho$ is in $\mathbf{U}_i$ for all $\rho \in [\![\Gamma]\!]$.*
(4) *If $\Gamma \vdash t : A : \Omega$ then $[\![A]\!]_\Gamma\ \rho$ is true for all $\rho \in [\![\Gamma]\!]$.*
(5) *If $\Gamma \vdash t : A : \mathcal{U}_i$ then $[\![t]\!]_\Gamma\ \rho$ is in $\mathsf{el}\ ([\![A]\!]_\Gamma\ \rho)$ for all $\rho \in [\![\Gamma]\!]$.*
(6) *If $\Gamma \vdash t \equiv u : A$ then $[\![t]\!]_\Gamma = [\![u]\!]_\Gamma$.*

$$
\begin{aligned}
[\![x]\!]_\Gamma\ \rho &:= \rho(x)\\
[\![\mathcal{U}_i]\!]_\Gamma\ \rho &:= \langle \mathbf{U}_i\ ;\ \mathrm{c}_\cup\ i\rangle\\
[\![\Omega]\!]_\Gamma\ \rho &:= \langle \Uparrow\ ;\ \mathrm{c}_\Omega\rangle\\
[\![\Pi^{\mathcal{U}_j,\mathcal{U}_k}(y:F).\,G]\!]_\Gamma\ \rho &:= \big\langle (x \in \mathsf{el}\ [\![F]\!]_\Gamma\ \rho) \to (\mathsf{el}\ [\![G]\!]_{\Gamma,x:F}\ \langle \rho\ ;\ x\rangle)\\
&\qquad ;\ \mathrm{c}_{\Pi\cup\cup}\ j\ k\ ([\![F]\!]_\Gamma\ \rho)\ (x \mapsto [\![G]\!]_{\Gamma,x:F}\ \langle \rho\ ;\ x\rangle)\big\rangle\\
[\![\Pi^{\Omega,\mathcal{U}_k}(y:F).\,G]\!]_\Gamma\ \rho &:= \big\langle (x \in \mathsf{val}\ [\![F]\!]_\Gamma\ \rho) \to (\mathsf{el}\ [\![G]\!]_{\Gamma,x:F}\ \langle \rho\ ;\ x\rangle)\\
&\qquad ;\ \mathrm{c}_{\Pi\Omega\cup}\ ([\![F]\!]_\Gamma\ \rho)\ (x \mapsto [\![G]\!]_{\Gamma,x:F}\ \langle \rho\ ;\ x\rangle)\big\rangle\\
[\![\Pi^{\mathcal{U}_j,\Omega}(y:F).\,G]\!]_\Gamma\ \rho &:= \forall x \in (\mathsf{el}\ [\![F]\!]_\Gamma\ \rho),\ [\![G]\!]_{\Gamma,x:F}\ \langle \rho\ ;\ x\rangle\\
[\![\Pi^{\Omega,\Omega}(y:F).\,G]\!]_\Gamma\ \rho &:= ([\![F]\!]_\Gamma\ \rho) \Rightarrow ([\![G]\!]_{\Gamma,x:F}\ \langle \rho\ ;\ *\rangle)\\
[\![\lambda(x:F).\,t]\!]_\Gamma\ \rho &:= x \mapsto ([\![t]\!]_{\Gamma,x:F}\ \langle \rho\ ;\ x\rangle)\\
[\![t\ u]\!]_\Gamma\ \rho &:= ([\![t]\!]_\Gamma\ \rho)\ ([\![u]\!]_\Gamma\ \rho)\\
[\![\mathbb{N}]\!]_\Gamma\ \rho &:= \langle \boldsymbol{\omega}\ ;\ \mathrm{c}_{\mathsf{emb}}\ \boldsymbol{\omega}\rangle\\
[\![0]\!]_\Gamma\ \rho &:= 0\\
[\![\mathrm{S}\ t]\!]_\Gamma\ \rho &:= \mathrm{S}\ ([\![t]\!]_\Gamma\ \rho)\\
[\![\mathbb{N}\text{-}\mathsf{elim}(P,t_0,t_S,n)]\!]_\Gamma\ \rho &:= \boldsymbol{\omega}\text{-}\mathsf{elim}(\mathsf{el} \circ ([\![P]\!]_\Gamma\ \rho), [\![t_0]\!]_\Gamma\ \rho, [\![t_S]\!]_\Gamma\ \rho, [\![n]\!]_\Gamma\ \rho)\\
[\![(y:F)\ \&\ G]\!]_\Gamma\ \rho &:= ([\![F]\!]_\Gamma\ \rho) \wedge ([\![G]\!]_{\Gamma,y:F}\ \langle \rho\ ;\ *\rangle)\\
[\![\langle t,u\rangle]\!]_\Gamma\ \rho &:= *\\
[\![\mathsf{fst}(t)]\!]_\Gamma\ \rho &:= *\\
[\![\mathsf{snd}(t)]\!]_\Gamma\ \rho &:= *\\
[\![\bot]\!]_\Gamma\ \rho &:= \emptyset \in \Uparrow\\
[\![\bot\text{-}\mathsf{elim}(A,t)]\!]_\Gamma\ \rho &:= \text{undefined}\\
[\![t \sim_A u]\!]_\Gamma\ \rho &:= [\![t]\!]_\Gamma\ \rho = [\![u]\!]_\Gamma\ \rho\\
[\![\mathsf{refl}(t)]\!]_\Gamma\ \rho &:= *\\
[\![\mathsf{transp}(F,t,G,u,t',e)]\!]_\Gamma\ \rho &:= *\\
[\![\mathsf{cast}(A,B,e,t)]\!]_\Gamma\ \rho &:= [\![t]\!]_\Gamma\ \rho\\
[\![\mathsf{castrefl}(A,t)]\!]_\Gamma\ \rho &:= *
\end{aligned}
$$

Fig. 11. Interpretation of $\mathrm{CC}^{\mathrm{obs}}$ in the Standard Model

Proof. By induction on the typing derivations, using Lemmas 5.1 and 5.2. □

5.5 Consequences of the Model

From the soundness theorem and the interpretation of $\bot$ as $\emptyset$, the consistency of $\mathrm{CC}^{\mathrm{obs}}$ is immediate:

Theorem 5.4 (Consistency). *There are no proofs of $\bot$ in the empty context.*

Furthermore, by inspecting the normal forms provided by the normalization theorem, we realize that cast is the only way to build a neutral term in the absence of variables. But having a stuck cast requires having an equality proof between two incompatible types, which contradicts consistency.

Theorem 5.5. *There are no neutral terms in the empty context.*

From there, we deduce the canonicity theorem: all integers reduce to standard integers in the empty context. Thus our model allows us to establish all the important meta-theoretical properties.

On the other hand, it seems difficult to measure to what extent our set-theoretic model presents $\mathrm{CC}^{\mathrm{obs}}$ as a good internal language for sets. Compared to the universe construction of Pujet and Tabareau [2022], we gain the property that every set belongs to a universe—meaning that theorems which quantify over $\mathcal{U}_i$ will apply to all sufficiently small sets. Of course, this does not prevent us from proving some theorems that are obviously false in classical mathematics, such as the injectivity of type formers. But we would argue that all such results are artifacts of the choice of encodings, and not meaningful mathematical statements.

6 $\mathrm{CC}^{\mathrm{obs}}$ WITH A PROOF-RELEVANT IMPREDICATIVE UNIVERSE HAS UNDECIDABLE TYPE-CHECKING

The system we have presented so far has an impredicative universe Ω of *proof-irrelevant* propositions, which contains the observational equality and the false proposition with the corresponding large elimination principles. In other words, Ω provides the user with a complete Heyting algebra of truth values, as is available in Coq and Lean.

However Ω is a bit weaker than the impredicative universe $\mathbb{P}\mathrm{rop}$ of the Calculus of Inductive Constructions. Indeed, $\mathbb{P}\mathrm{rop}$ allows large elimination for all the *sub-singleton* inductive propositions, *i.e.* the propositions that have at most one constructor and whose constructor arguments all have sort $\mathbb{P}\mathrm{rop}$. This includes equality, but also the accessibility predicate Acc from which we can derive some constructive choice principles (as discussed in [Forster 2021]) that may come in handy for mathematics. However, the price to pay for this additional logical power is that proof terms in $\mathbb{P}\mathrm{rop}$ are *proof-relevant*, and need to be reduced.

Thus one may wonder if we can add $\mathbb{P}\mathrm{rop}$ to $\mathrm{CC}^{\mathrm{obs}}$ alongside the universe of strict propositions, using the same computation rules as for the proof-relevant fragment of $\mathrm{CC}^{\mathrm{obs}}$ so that it can provide us with more logical power while still enjoying extensionality principles. We need to be careful though, as past experiments of adding a proof-irrelevant equality to Coq that have been mentioned in [Gilbert et al. 2019] have resulted in the breakage of normalization [Abel and Coquand 2020].

The argument of Abel and Coquand relies on the fact that equality behaves like the usual Martin-Löf Identity Type, which is an inductive type that computes *via* the J-eliminator. In $\mathrm{CC}^{\mathrm{obs}}$, elimination of equality is done using the cast operator which is quite different, as it computes by comparing the weak-head normal forms of the source and target types. Unfortunately, it turns out that $\mathrm{CC}^{\mathrm{obs}}$ is not compatible with $\mathbb{P}\mathrm{rop}$ either. In fact, we show in this section that it is possible to use a modified version of Abel and Coquand's argument to show undecidability of type-checking for open terms in presence of $\mathbb{P}\mathrm{rop}$, both for $\mathrm{CC}^{\mathrm{obs}}$ and the theory of Abel and Coquand [2020].

6.1 Outline

The idea of the proof is rather simple: on the one hand, impredicativity makes it possible to give a type to the term

$$\Delta_f := \lambda x.\ f\ (x\ x)$$

for any well-typed function f of type $(\mathbb{N}_{\mathbb{P}} \to \mathbb{N}_{\mathbb{P}}) \to (\mathbb{N}_{\mathbb{P}} \to \mathbb{N}_{\mathbb{P}})$, which is one half of the fixed point combinator $Y_f := \Delta_f\ \Delta_f$. Here $\mathbb{N}_{\mathbb{P}}$ denotes the inductive natural numbers in $\mathbb{P}\mathrm{rop}$. On the other hand, cast can be used to convert between any two types in an inconsistent context, in a way that does not block reduction. This allows us to apply Δ_f to itself (up to a cast) and obtain an approximation of Y_f. Then, using these pseudo-fixed points, we can build a term that loops through the iterates of any integer function:

Theorem 6.1. *Let g be any closed term of type $\mathbb{N}_{\mathbb{P}} \to \mathbb{N}_{\mathbb{P}}$. In an inconsistent context, one can define a term dec_g of type $\mathbb{N}_{\mathbb{P}}$ which is convertible to 0 if and only if there is a positive integer n such that $g^n\ 1 = 0$, and diverges otherwise.*

Of course, the existence of such an integer is not decidable in general, thus conversion and typing are undecidable for $\mathrm{CC}^{\mathrm{obs}} + \mathbb{P}\mathrm{rop}$. As an aside, note that this pseudo-fixed point operator can also be constructed in Ω if we add proof-irrelevant natural numbers. But it does not cause any issue, as we have a simple way to check for convertibility of proof-irrelevant terms: having the same type is sufficient! For this reason, we never perform reduction in the proof-irrelevant layer, and need not to worry about these potentially divergent terms.

6.2 Definition of an Approximate Fixed Point Combinator

Let f be a closed function of type $(\mathbb{N}_\mathbb{P} \to \mathbb{N}_\mathbb{P}) \to (\mathbb{N}_\mathbb{P} \to \mathbb{N}_\mathbb{P})$. In an inconsistent context, we can derive a term

$$e : \Pi(X\ Y : \mathbb{P}\mathrm{rop}).\ X \sim Y$$

that allows us to freely cast between any two propositions. We can use it to derive our approximate fixpoint combinator by defining

$$\begin{array}{lcll} \bot_\mathbb{P} & : & \mathbb{P}\mathrm{rop} & := \Pi(X : \mathbb{P}\mathrm{rop}).\ X \\ \Delta_f & : & \bot_\mathbb{P} \to \mathbb{N}_\mathbb{P} \to \mathbb{N}_\mathbb{P} & := \lambda(x : \bot_\mathbb{P}).\ f\ (x\ (\bot_\mathbb{P} \to \mathbb{N}_\mathbb{P} \to \mathbb{N}_\mathbb{P})\ x) \\ \Delta'_f & : & \bot_\mathbb{P} & := \lambda(X : \mathbb{P}\mathrm{rop}).\ \mathrm{cast}(\bot_\mathbb{P} \to \mathbb{N}_\mathbb{P} \to \mathbb{N}_\mathbb{P}, X, e\ (\bot_\mathbb{P} \to \mathbb{N}_\mathbb{P} \to \mathbb{N}_\mathbb{P})\ X, \Delta_f) \\ Y_f & : & \mathbb{N}_\mathbb{P} \to \mathbb{N}_\mathbb{P} & := \Delta_f\ \Delta'_f \end{array}$$

Technically, it turns out that Y_f does not exactly behave like a fixpoint combinator because of additional casts appearing during the reduction: a tedious but straightforward computation in $\mathrm{CC}^{\mathrm{obs}}$ shows that

$$\begin{aligned} Y_f &\equiv \Delta_f\ \Delta'_f \\ &\Rightarrow^* f\ (\alpha\ (\Delta_f\ (\beta\ \Delta'_f))) \\ &\Rightarrow^* f\ (\alpha\ (f\ (\alpha^2\ (\Delta_f\ (\beta^3\ \Delta'_f))))) \\ &\Rightarrow^* f\ (\alpha\ (f\ (\alpha^2\ (f\ (\alpha^4\ (\Delta_f\ (\beta^7\ \Delta'_f))))))) \\ &\Rightarrow^* \dots \end{aligned}$$

where $\alpha := \lambda(x : \mathbb{N}_\mathbb{P} \to \mathbb{N}_\mathbb{P}).\ \mathrm{cast}(\mathbb{N}_\mathbb{P} \to \mathbb{N}_\mathbb{P}, \mathbb{N}_\mathbb{P} \to \mathbb{N}_\mathbb{P}, e_2, x)$, $\beta := \lambda(x : \bot_\mathbb{P}).\ \mathrm{cast}(\bot_\mathbb{P}, \bot_\mathbb{P}, e_1, x)$ and e_1 and e_2 are equalities given respectively by the first and second projection of the equality $e\ (\bot_\mathbb{P} \to \mathbb{N}_\mathbb{P} \to \mathbb{N}_\mathbb{P})\ (\bot_\mathbb{P} \to \mathbb{N}_\mathbb{P} \to \mathbb{N}_\mathbb{P})$ obtained during cast reduction.

However, this behavior is sufficient for our purposes, since all the α disappear when the function is applied to an actual integer. Therefore, given a closed function g of type $\mathbb{N}_\mathbb{P} \to \mathbb{N}_\mathbb{P}$ that terminates on all integer inputs, we instantiate f with

$$\lambda(p : \mathbb{N}_\mathbb{P} \to \mathbb{N}_\mathbb{P}).\ \lambda(n : \mathbb{N}_\mathbb{P}).\ \mathbb{N}\text{–elim}(\mathbb{N}_\mathbb{P}, 0, \lambda(m : \mathbb{N}_\mathbb{P}).\ p\ (g\ m), n)$$

and we obtain that $Y_f\ 1$ reduces to 0 if there exists a n such that $g^n\ 1 = 0$, and diverges (for any reduction strategy) otherwise.

By replacing cast with the transport operator derived from the elimination principle of the inductive equality, the same result applies to the theory of Abel and Coquand [2020], even though the reduction pattern of Y_f is slightly different.

6.3 Conversion is Undecidable

$\mathrm{CC}^{\mathrm{obs}}$ is expressive enough to encode Turing machines so that knowing whether there exists a n such that $g^n\ 1 = 0$ is undecidable. To complete the proof of undecidability of conversion, we still need to show that a term of type $\mathbb{N}_\mathbb{P}$ that diverges for all reduction strategies is not convertible to 0. In other words, we need to show that conversion is not degenerate.

To establish this, we use a lemma of Geuvers, adapted to the Calculus of Inductive Constructions by Werner [1994, lemma 2.19]. The lemma corresponds to a weak form of confluence: any well-typed term t that is $\beta\eta\iota$-equal to a weak head normal form t' actually reduces to an η-expanded form of t' for untyped $\beta\iota'$ reduction, a slightly modified version of untyped $\beta\iota$ reduction. If we take $t' = 0$ then we obtain that a well-typed term t that is $\beta\eta\iota$-equal to 0 has a normal form, and thus cannot be divergent.

In order to apply this lemma to $\mathrm{CC}^{\mathrm{obs}}$, we need to extend the proof to strict propositions and type-casting operators. The extension is not too difficult: strict propositions do not contribute to the reduction behavior, and type-casting is very similar to an eliminator of an inductive type.

Again, a similar proof can be done in the theory of Abel and Coquand [2020].

7 CONCLUSION AND RELATED WORK

In this paper, we presented a way to reconcile proof-irrelevant impredicativity with UIP. Our work was born from the study of observational type theories, originating from the study of the setoid model of type theory [Altenkirch 1999; Hofmann 1995], and having seen renewed interest by the community in recent years [Altenkirch et al. 2019; Pujet and Tabareau 2022; Sterling et al. 2019].

This detour lead us rather far from the system that sparked the initial conundrum, which was a straightforward extension of MLTT with definitional UIP. The system in question features an especially interesting computation rule, which basically amounts to having a type-cast operator that reduces whenever the source and target types are convertible instead of computing by case analysis on the head constructor of said types. This computational behaviour trades the extensionality principles of the observational equality for a J operator that always reduces on reflexivity (J-on-refl), even with open terms. Pujet and Tabareau [2022] explain how to define a *proof-relevant* equality that recovers the J-on-refl computation rule in $\mathrm{TT}^{\mathrm{obs}}$, but it would be interesting to investigate the J-on-refl rule for the proof-irrelevant observational equality, so that we get the best of both worlds. This was already suggested by Allais et al. [2013] in the context of Observational Type Theory. Since we pinpointed the decidablity issues as stemming from computational impredicativity, we expect that J-on-refl would not be a problem in a theory with proof-irrelevant impredicativity such as $\mathrm{CC}^{\mathrm{obs}}$.

The set-theoretic model of Section 5 fulfills everything we could expect of a "standard" model, and ensures us that the theorems of $\mathrm{CC}^{\mathrm{obs}}$ cannot deviate too much from the mathematical canon. However, $\mathrm{CC}^{\mathrm{obs}}$ is a *constructive* type theory, and as such we can hope that it has natural semantics in a broader class of models: Grothendieck toposes. Indeed, $\mathrm{CC}^{\mathrm{obs}}$ is characterized by its extensional equality and an impredicative universe of propositions, two features that seem to be a perfect match for topos theory. This was first noticed by Gratzer [2022], who remarked that the model construction that we presented in Section 5 is sufficiently general to work in any Grothendieck topos. This makes $\mathrm{CC}^{\mathrm{obs}}$ a reasonably good internal language for toposes, but it still lacks one important feature that is available in all toposes: the principle of unique choice. This principle, also known as *function comprehension*, states that given any functional relation $R : A \to B \to \Omega$ we can construct a choice function for R

$$\Pi(x : A).\, \exists!(y : B)\ .\ R\ x\ y \quad \to \quad \Sigma(f : A \to B).\, \Pi(x : A).\, R\ x\ (f\ x).$$

using the Σ-types from Pujet and Tabareau [2022] and assuming that the unique existence quantifier is defined with the usual impredicative encoding. Unfortunately, this principle is impossible to derive in $\mathrm{CC}^{\mathrm{obs}}$.

REFERENCES

Andreas Abel, Klaus Aehlig, and Peter Dybjer. 2007. Normalization by evaluation for Martin-Löf type theory with one universe. *Electronic Notes in Theoretical Computer Science* 173 (2007), 17–39.

Andreas Abel and Thierry Coquand. 2005. Untyped Algorithmic Equality for Martin-Löf's Logical Framework with Surjective Pairs. In *Typed Lambda Calculi and Applications*, Paweł Urzyczyn (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 23–38.

Andreas Abel and Thierry Coquand. 2020. Failure of Normalization in Impredicative Type Theory with Proof-Irrelevant Propositional Equality. *Logical Methods in Computer Science* Volume 16, Issue 2 (June 2020). https://doi.org/10.23638/LMCS-16(2:14)2020

Andreas Abel, Joakim Öhman, and Andrea Vezzosi. 2018. Decidability of Conversion for Type Theory in Type Theory. *Proceedings of the ACM on Programming Languages* 2, POPL, Article 23 (Jan. 2018), 29 pages. https://doi.org/10.1145/3158111

Guillaume Allais, Conor McBride, and Pierre Boutillier. 2013. New Equations for Neutral Terms: A Sound and Complete Decision Procedure, Formalized. In *Proceedings of the 2013 ACM SIGPLAN Workshop on Dependently-typed Programming* (Boston, Massachusetts, USA) *(DTP '13)*. ACM, New York, NY, USA, 13–24. https://doi.org/10.1145/2502409.2502411

T. Altenkirch. 1999. Extensional equality in intensional type theory. In *Proceedings. 14th Symposium on Logic in Computer Science (Cat. No. PR00158)*. 412–420. https://doi.org/10.1109/LICS.1999.782636

Thorsten Altenkirch, Simon Boulier, Ambrus Kaposi, and Nicolas Tabareau. 2019. Setoid type theory - a syntactic translation. In *MPC 2019 - 13th International Conference on Mathematics of Program Construction (LNCS, Vol. 11825)*. Springer, 155–196. https://doi.org/10.1007/978-3-030-33636-3_7

Thorsten Altenkirch, Conor McBride, and Wouter Swierstra. 2007. Observational equality, now!. In *Proceedings of the Workshop on Programming Languages meets Program Verification (PLPV 2007)*. 57–68. https://doi.org/10.1145/1292597.1292608

HP Barendregt. 1993. Lambda calculi with types. In *Handbook of logic in computer science (vol. 2) background: computational structures*. 117–309.

Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. 2015. Cubical Type Theory: a constructive interpretation of the univalence axiom. In *21st International Conference on Types for Proofs and Programs (21st International Conference on Types for Proofs and Programs, 69)*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Tallinn, Estonia, 262. https://doi.org/10.4230/LIPIcs.TYPES.2015.5

Thierry Coquand and Gérard Huet. 1988. The calculus of constructions. *Information and Computation* 76, 2 (1988), 95–120. https://doi.org/10.1016/0890-5401(88)90005-3

Yannick Forster. 2021. Church's Thesis and Related Axioms in Coq's Type Theory. In *29th EACSL Annual Conference on Computer Science Logic (CSL 2021) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 183)*, Christel Baier and Jean Goubault-Larrecq (Eds.). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 21:1–21:19. https://doi.org/10.4230/LIPIcs.CSL.2021.21

Gaëtan Gilbert, Jesper Cockx, Matthieu Sozeau, and Nicolas Tabareau. 2019. Definitional Proof-Irrelevance without K. *Proceedings of the ACM on Programming Languages* 3 (Jan. 2019), 1–28. https://doi.org/10.1145/3290316

Jean-Yves Girard. 1972. Interprétation fonctionnelle et élimination des coupures dans l'arithmétique d'ordre supérieur. (1972). Thèse de Doctorat d'État, Université de Paris VII.

Daniel Gratzer. 2022. An inductive-recursive universe generic for small families. https://doi.org/10.48550/ARXIV.2202.05529

Peter Hancock, Conor McBride, Neil Ghani, Lorenzo Malatesta, and Thorsten Altenkirch. 2013. Small Induction Recursion. In *Typed Lambda Calculi and Applications*, Masahito Hasegawa (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 156–172.

Martin Hofmann. 1995. *Extensional concepts in intensional type theory.* Ph. D. Dissertation. University of Edinburgh.

Chung-Kil Hur, Georg Neis, Derek Dreyer, and Viktor Vafeiadis. 2013. The Power of Parameterization in Coinductive Proof. In *Proceedings of the 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (Rome, Italy) *(POPL '13)*. Association for Computing Machinery, New York, NY, USA, 193–206. https://doi.org/10.1145/2429069.2429093

Antonius J. C. Hurkens. 1995. A Simplification of Girard's Paradox. In *Proceedings of the Second International Conference on Typed Lambda Calculi and Applications (TLCA '95)*. Springer-Verlag, Berlin, Heidelberg, 266–278.

Per Martin-Löf. 1975. An Intuitionistic Theory of Types: Predicative Part. In *Logic Colloquium '73*, H.E. Rose and J.C. Shepherdson (Eds.). Studies in Logic and the Foundations of Mathematics, Vol. 80. Elsevier, 73 – 118. https://doi.org/10.1016/S0049-237X(08)71945-1

Loïc Pujet and Nicolas Tabareau. 2022. Observational Equality: Now For Good. *Proceedings of the ACM on Programming Languages* 6, POPL (Jan. 2022), 1–29. https://doi.org/10.1145/3498693

Jonathan Sterling, Carlo Angiuli, and Daniel Gratzer. 2019. Cubical Syntax for Reflection-Free Extensional Equality. In *4th International Conference on Formal Structures for Computation and Deduction (FSCD 2019) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 131)*, Herman Geuvers (Ed.). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 31:1–31:25. https://doi.org/10.4230/LIPIcs.FSCD.2019.31

Benjamin Werner. 1994. *Une Théorie des Constructions Inductives.* Theses. Université Paris-Diderot - Paris VII. https://tel.archives-ouvertes.fr/tel-00196524

Received 2022-07-07; accepted 2022-11-07