Cubical Syntax for Reflection-Free Extensional Equality

Jonathan Sterling
Carnegie Mellon University
http://cs.cmu.edu/~jmsterli
jmsterli@cs.cmu.edu

Carlo Angiuli
Carnegie Mellon University
http://cs.cmu.edu/~cangiuli
cangiuli@cs.cmu.edu

Daniel Gratzer
Aarhus University
http://jozefg.github.io
gratzer@cs.au.dk

Abstract

We contribute XTT, a cubical reconstruction of Observational Type Theory [5] which extends Martin-Löf's intensional type theory with a *dependent equality type* that enjoys function extensionality and a judgmental version of the *unicity of identity proofs* principle (UIP): any two elements of the same equality type are judgmentally equal. Moreover, we conjecture that the typing relation can be decided in a practical way. In this paper, we establish an algebraic canonicity theorem using a novel extension of the *logical families* or *categorical gluing* argument inspired by Coquand and Shulman [20, 36]: every closed element of boolean type is derivably equal to either true or false.

2012 ACM Subject Classification Theory of computation → Type theory; Theory of computation → Algebraic semantics; Theory of computation → Denotational semantics

Keywords and phrases Dependent type theory, extensional equality, cubical type theory, categorical gluing, canonicity

Funding The authors gratefully acknowledge the support of the Air Force Office of Scientific Research through MURI grant FA9550-15-1-0053. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the AFOSR.

Acknowledgements We thank Lars Birkedal, Evan Cavallo, David Thrane Christiansen, Thierry Coquand, Kuen-Bang Hou (Favonia), Marcelo Fiore, Jonas Frey, Krzysztof Kapulkin, András Kovács, Dan Licata, Conor McBride, Darin Morrison, Anders Mörtberg, Michael Shulman, Bas Spitters, and Thomas Streicher for helpful conversations about extensional equality, algebraic type theory, and categorical gluing. We especially thank Robert Harper for valuable conversations throughout the development of this work. We also thank Paul Taylor for his diagrams package, which we have used to typeset the commutative diagrams in this paper.

1 Introduction

The past fifty years of constructive type theory can be summed up as the search for a scientific understanding of *equality*, punctuated by moments of qualitative change in our perception of the boundary between semantics (actual construction) and syntax (proof theory) from a type-theoretic point of view. Computation is critical to both the semantics and syntax of type theory—from Martin-Löf's meaning explanations [33], supplying type theory with its direct semantics and intuitionistic grounding, to syntactic properties such as closed and open *canonicity* which establish computation as the indispensible method for deriving equations.

For too long, a limiting perspective on extensional type theory has prevailed, casting it as a particular syntactic artifact (for instance, the formalism obtained by stripping of their meaning the rules which incidentally appear in Martin-Löf's monograph [33]), a formal system which enjoys precious few desirable syntactic properties and is distinguished primarily by its *equality reflection* rule:

$$\frac{\Gamma \vdash P \in \mathsf{Eq}_A(M, N)}{\Gamma \vdash M = N \in A}\ \textsc{reflection}$$

We insist on the contrary that the conceptual importance of extensional type theory lies not in the specific choice of syntactic presentation (historically, via equality reflection), but rather in the *semantic* characteristics of its equality connective, which are invariant under choice of syntax. The specifics of how such an equality construct is presented syntactically are entirely negotiable (the internal language of a doctrine is determined only up to equivalence), and therefore has an empirical component.

1.1 Syntax-invariant criteria for extensional equality

Extensional equality should enjoy at least the following syntax-invariant properties:

1. Equality must be reflexive, symmetric and transitive.
2. Equality must support *coherent coercion*: if $\mathsf{Eq}_A(M, N)$ and P is a family of types over A, then we must have an isomorphism $P(M) \to P(N)$ subject to some natural equations.
3. Proofs of equality should be *unique*: if $P, Q : \mathsf{Eq}_A(M, N)$, then we must *at least* have $\mathsf{Eq}_{\mathsf{Eq}_A(M,N)}(P, Q)$.
4. The equality type for compound types, such as $\mathsf{Eq}_{A\to B}(f, g)$, should be isomorphic to the corresponding compound type of equality types, sc. $(x : A) \to \mathsf{Eq}_B(f(x), g(x))$.

The first two criteria are definitive of equality generally, whereas the latter two are characteristic of *extensional* notions of equality. The uniqueness criterion distinguishes proper equality from other notions, such as paths [41]; within a particular syntactic presentation, one might arrange for the uniqueness to hold judgmentally, or only up to a proof. In this paper, we arrange for XTT's equality types to have a judgmental uniqueness principle.

The last criterion—that the equality connective should commute in an appropriate way with the other connectives of type theory—is really the demand that the equality connective should *in fact* internalize the judgmental equality, in exactly the same sense that the dependent function connective internalizes the hypothetical judgment. For instance, the principle of function extensionality internalizes the judgmental extensionality principle of the function connective (derivable from rules η, ξ, β):

$$\frac{\Gamma, x : A \vdash f(x) = g(x) : B}{\Gamma \vdash f = g : A \to B} \quad \textit{internalized as} \quad \frac{(x : A) \to \mathsf{Eq}_B(f(x), g(x))}{\mathsf{Eq}_{A\to B}(f, g)}$$

The *equality reflection* rule is one way to achieve all these properties simultaneously in a particular syntactic presentation of equality, but it is by no means the only way.

1.2 Extensional equality via equality reflection

The earliest type-theoretic proof assistants employed the equality reflection rule (or equivalent formulations) in order to achieve syntax for extensional equality, a method most famously represented by Nuprl [18] and its descendents, including **RedPRL** [8]. The Nuprl-style formalisms act as a "window on the truth" for a *single* intended semantics inspired by Martin-Löf's computational meaning explanations [2]; semantic justification in the computational ontology is the *only* consideration when extending the Nuprl formalism with a new rule, in contrast to other traditions in which global properties (e.g. admissibility of structural rules, decidability of typing, interpretability in multiple models, etc.) are treated as definitive.

Rather than using *type checking*, proof assistants in this style support interactive development of typing derivations using tactics and partial decision procedures. A notable aspect of the Nuprl family is that their formal sequents range not over typed terms (proofs), but over untyped raw terms (realizers); a consequence is that during the proof process, one must repeatedly establish numerous *type functionality* subgoals, which restore the information that is lost when passing from a proof to a realizer. To mitigate the corresponding blow-up in proof size, Nuprl relies heavily on untyped computational reasoning via pointwise functionality, a non-standard but consistent semantics for dependently typed sequents which has some surprising consequences, such as refuting the principle of *dependent cut* [31].

Another approach to implementing type theory with equality reflection is exemplified in the experimental Andromeda proof assistant [12], in which proofs are also built interactively using tactics, but judgments range over abstract proof derivations rather than realizers. This approach mitigates to some degree the practical problems caused by erasing information prematurely, and also enables interpretation into a broad class of semantic models.

Although Nuprl/**RedPRL** and Andromeda illustrate that techniques beyond mere type checking are profitable to explore, the authors' experiences building and using **RedPRL** for concrete formalization of mathematics underscored the benefits of having a practical algorithm to check types, particularly in the setting of cubical type theory (Section 1.6), whose higher-dimensional structure significantly reduces the applicability of Nuprl-style untyped reasoning. In particular, whereas it is possible to treat *all* β-rules and many η-rules in non-cubical type theory as untyped rewrites, this method is unsound for the cubical account of higher inductive types and univalence [9]; consequently, in **RedPRL** many β/η rewrites must emit auxiliary proof obligations. Synthesizing these experiences and challenges led to the creation of the `redtt` proof assistant for Cartesian cubical type theory [7].

1.3 Equality in intensional type theory

Martin-Löf's Intensional Type Theory [32, 35] underapproximates equality via its *identity type*, characterized by rules like the following:

FORMATION
$$\frac{\Gamma \vdash A\ \mathit{type} \qquad \Gamma \vdash M : A \qquad \Gamma \vdash N : A}{\Gamma \vdash \mathsf{Id}_A(M, N)\ \mathit{type}}$$

INTRODUCTION
$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \mathsf{refl}_A(M) : \mathsf{Id}_A(M, M)}$$

ELIMINATION
$$\frac{\begin{array}{c}\Gamma, x : A, y : A, z : \mathsf{Id}_A(x, y) \vdash C(x, y, z)\ \mathsf{type} \\ \Gamma \vdash P : \mathsf{Id}_A(M, N) \qquad \Gamma, x : A \vdash Q : C(x, x, \mathsf{refl}_A(x))\end{array}}{\Gamma \vdash \mathsf{J}_{x,y,z.C}(P; x.Q) : C(M, N, P)} \qquad \ldots$$

Symmetry, transitivity and coercion follow from the elimination rule of the identity type. However, the unicity criterion is independent of Intensional Type Theory [27]; likewise, there are sufficiently intensional models of the identity type to refute the principle of function extensionality [39]. Therefore, we can say definitively that Martin-Löf's identity type failed to internalize extensional equality in the sense that we demanded in Section 1.1.

1.4 Setoids and internal model constructions

A standard technique for avoiding the deficiencies of the identity type in Intensional Type Theory is the *setoid construction* [25], an exact completion which glues an equivalence relation $=_A$ onto each type $|A|$ in the spirit of Bishop [13]. When using setoids, a function $A \to B$ consists of a type-theoretic function $f : |A| \to |B|$ together with a proof that it preserves the equivalence relation, $f_= : (x, y : |A|) \to x =_A y \to f(x) =_B f(y)$; a *dependent setoid* (family of setoids) is a type-theoretic family equipped with a coherent coercion operator.

Setoids are a discipline for expressing internally precisely the extrinsic properties required for constructions to be extensional (compatible with equality); these extra proof obligations must be satisfied in parallel with constructions at every turn. The state of affairs for setoids is essentially analogous to that of proof assistants with equality reflection, in which type functionality subgoals play a similar role to auxiliary paperwork generated by setoids.

Paradoxically, however, every construction in ordinary Intensional Type Theory is automatically extensional in this sense. A solution to the problem of equality in type theory should, unlike setoids, take advantage of the fact that type theory is already restricted to extensional constructions, adding to it only enough language to refer to equality internally. This is the approach taken by both Observational Type Theory and XTT.

1.5 Observational Type Theory

The first systematic solution to the problem of syntax for extensional equality without equality reflection was Observational Type Theory (OTT) [4, 5], which built on early work by Altenkirch and McBride [3, 34]. The central idea of OTT is to work with a *closed* universe of types, defining by recursion for each pair of types A, B a type $\mathsf{Eq}(A, B)$ of proofs that A and B are equal, and for each pair of elements $M : A$ and $N : B$, a type of proofs $\mathsf{Eq}_{A,B}(M, N)$ that M and N are (heterogeneously) equal. Finally, one defines "generic programs" by recursion on type structure which calculate coercions and coherences along proofs of equality.

One can think of OTT as equipping the semantic setoid construction with a direct-style type-theoretic language, and adding to it closed, inductively defined universes of types. The heterogeneous equality of OTT, initially a simplifying measure adopted from McBride's thesis [34], is an early precursor of the *dependent paths* which appear in Homotopy Type Theory [41] and Cubical Type Theory [17, 6, 9], as well as XTT.

Recently, McBride and his collaborators have made progress toward a cubical version of OTT, using a different cube category and coercion structure, in which one coerces only from 0 to 1, and obtains fillers using an affine rescaling operation [16].

1.6 Cubical Type Theory

In a rather different line of research, Voevodsky showed that Intensional Type Theory is compatible with a *univalence axiom* yielding an element of $\mathsf{Id}_{\mathcal{U}}(A, B)$ for every equivalence (coherent isomorphism) between types A, B [30]. The resulting notion of identity captures not extensional equality but rather a notion of *path* analogous to that of topological spaces [41]. However, Intensional Type Theory extended with univalence lacks *canonicity*, because identity elimination computes only on refl and not on proofs constructed by univalence.

Since then, *cubical type theories* have been developed to validate univalence without disrupting canonicity [17, 9]. These type theories extend Martin-Löf's type theory with an abstract interval, maps out of which represent paths; the interval has both abstract elements, represented by a new sort of *dimension* variable i, and constant endpoints $0, 1$. Coercions are an instance of *Kan structure* governed directly by the structure of paths between types, which are nothing more than types dependent on an additional dimension variable.

There are currently two major formulations of cubical type theory. De Morgan cubical type theory [17] equips the interval with negation and binary connection (minimum and maximum) operations. Cartesian cubical type theory [6, 9], the closest relative of XTT, has no additional structure on the interval, but equips types with a much stronger notion of coercion generalizing the one described in Section 2.1.1.

1.7 Our contribution: XTT

We contribute XTT, a new type theory that supports extensional equality without equality reflection, using ideas from cubical type theory [17, 6, 9]. In particular, we obtain a compositional account of propositional equality which internalizes judgmental equality at every type (including, for instance, function extensionality). Moreover, XTT supports a *judgmental* version of the unicity criterion for equality discussed in Section 1.1—when $P, Q : \mathsf{Eq}_A(M, N)$, we have $P = Q$ judgmentally—allowing us to substantially simplify our Kan operations (Section 2.1.2). Finally, XTT is closed under a cumulative hierarchy[1] of closed universes à la Russell. We hope to integrate XTT into the redtt cubical proof assistant [7] as an implementation of extensional equality in the style of two-level type theory [9].

A common thread that runs through the XTT formalism is the decomposition of constructs from OTT into more modular, *judgmental* principles. For instance, rather than defining equality separately at every type and entangling the connectives, equality is defined once and for all using the interval; likewise, rather than ensuring that equality proofs are unique through brute force, we obtain this using a structural rule which does not mention the equality type.

By first developing the model theory of XTT in an algebraic way (Section 3), we then prove a canonicity theorem for the *initial* model of XTT (Section 3.2): any closed term of boolean type is equal to either true or false. This result is obtained using a novel extension of the categorical gluing technique described by Coquand and Shulman [20, 36]. Canonicity expresses a form of "computational adequacy"—in essence, that the equational theory of XTT suffices to derive any equation which ought to hold by (closed) computation—and is one of many syntactical considerations that experience has shown to be correlated to usability.

2 Programming and proving in XTT

Like other cubical type theories, the XTT language extends Martin-Löf's type theory with a new sort of variable i ranging over an abstract interval with inhabitants 0 and 1; we call an element r of the interval a *dimension*, and we write ε to range over a constant dimension 0 or 1. Cubical type theories like XTT also use a special kind of hypothesis to constrain the values of dimensions: when r and s are dimensions, then $r = s$ is a *constraint*. In XTT, a single context Ψ cube_+ accounts for both dimension variables (Ψ, i) and constraints $(\Psi, r = s)$. Dimensions can be substituted for dimension variables, an operation written $M\langle r/i\rangle$.

Finally, ordinary type-theoretic assumptions $x : A$ are kept in a context Γ that depends on Ψ. In XTT, a full context is therefore written $\Psi \mid \Gamma$. The meaning of a judgment at context $(\Psi, i = r)$ is completely determined by its instance under the substitution r/i. Under the

[1] As in previous work [37], we employ an *algebraic* version of cumulativity which does not require subtyping.

(cubes)	Ψ, Φ	$::=$	$\cdot \mid \Psi, i \mid \Psi, \xi$
(contexts)	Γ, Δ	$::=$	$\cdot \mid \Gamma, x : A$
(dimensions)	r, s	$::=$	$i \mid \varepsilon$
(constant dims.)	ε	$::=$	$0 \mid 1$
(constraints)	ξ	$::=$	$r = r'$
(universe levels)	k, l	$::=$	$n \quad (n \in \mathbb{N})$
(types)	A, B	$::=$	$M \mid (x : A) \to B \mid (x : A) \times B \mid \mathsf{Eq}_{i.A}(M, N) \mid \Uparrow^l_k A \mid \mathcal{U}_k \mid \mathsf{bool}$
(terms)	M, N	$::=$	$x \mid A \mid \lambda x.M \mid \mathsf{app}_{x:A.B}(M, N) \mid \langle M, N\rangle \mid \mathsf{fst}_{x:A.B}(M) \mid \mathsf{snd}_{x:A.B}(M) \mid$ $\lambda i.M \mid \mathsf{app}_{i.A}(M, r) \mid \mathsf{true} \mid \mathsf{false} \mid \mathsf{if}_{x.A}(M; N_0, N_1) \mid$ $[i.A] \downarrow^r_{r'} M \mid A \downarrow^r_{r'} M \; [s \text{ with } 0 \hookrightarrow j.N_0 \mid 1 \hookrightarrow j.N_1]$

Figure 1 A summary of the raw syntax of XTT. As a matter of top-level notation, we omit annotations that can be inferred from context, writing $M(N)$ for $\mathsf{app}_{x:A.B}(M, N)$.

false constraint $0 = 1$, all judgments hold; the resulting collapse of the typing judgment and the judgmental equality does not disrupt any important metatheoretic properties, because the theory of dimensions is decidable.

The general typehood judgment $\Psi \mid \Gamma \vdash A \ \mathit{type}_k$ means that A is a type of universe level k in context Γ over the cube Ψ; note that this judgment presupposes the well-formedness of Ψ, Γ. Likewise, the element typing judgment $\Psi \mid \Gamma \vdash M : A$ means that M is an element of the type A over Ψ as above; this form of judgment presupposes the well-formedness of A and thence Ψ, Γ. We also have typed judgmental equality $\Psi \mid \Gamma \vdash A = B \ \mathit{type}_k$ and $\Psi \mid \Gamma \vdash M = N : A$, which presuppose the well-formedness of *all* their constituents.

Dependent equality types

XTT extends Martin-Löf type theory with *dependent equality types* $\mathsf{Eq}_{i.A}(N_0, N_1)$ when $\Psi, i \mid \Gamma \vdash A \ \mathit{type}_k$ and $\Psi \mid \Gamma \vdash N_0 : A\langle 0/i\rangle$ and $\Psi \mid \Gamma \vdash N_1 : A\langle 1/i\rangle$. Geometrically, elements of this type are *lines* or *paths* in the type A ranging over dimension i, with left endpoint N_0 and right endpoint N_1.[2] This type captures internally the equality of N_0 and N_1; dependency of A on the dimension i is in essence a cubical reconstruction of heterogeneous equality, albeit with different properties from the version invented by McBride in his thesis [34].

An element of the equality type $\mathsf{Eq}_{i.A}(N_0, N_1)$ is formed by the dimension λ-abstraction $\lambda i.M$, requiring that M is an element of A in the extended context, and that N_0, N_1 are the left and right sides of M respectively. Proofs P of equality are eliminated by dimension application, $P(r)$, and are subject to β, η, ξ rules analogous to those for function types. Finally, we have $P(\varepsilon) = N_\varepsilon$ always, extending Gentzen's principle of inversion to the side condition that we placed on M. More formally:

$$\frac{\Psi, i \mid \Gamma \vdash M : A \qquad \overrightarrow{\Psi, i = \varepsilon \mid \Gamma \vdash M = N_\varepsilon : A}}{\Psi \mid \Gamma \vdash \lambda i.M : \mathsf{Eq}_{i.A}(N_0, N_1)} \qquad \frac{\Psi \mid r \ \mathit{dim} \qquad \Psi \mid \Gamma \vdash M : \mathsf{Eq}_{i.A}(N_0, N_1)}{\Psi \mid \Gamma \vdash M(r) : A\langle r/i\rangle}$$

$$\frac{\Psi \mid \Gamma \vdash M : \mathsf{Eq}_{i.A}(N_0, N_1)}{\Psi \mid \Gamma \vdash M(\varepsilon) = N_\varepsilon : A\langle \varepsilon/i\rangle} \qquad \frac{\Psi \mid \Gamma \vdash M : \mathsf{Eq}_{i.A}(N_0, N_1)}{\Psi \mid \Gamma \vdash M = \lambda i.M(i) : \mathsf{Eq}_{i.A}(N_0, N_1)}$$

[2] Our dependent equality types are locally the same as dependent path types $\mathsf{Path}_{i.A}(N_0, N_1)$ from cubical type theories; however, we have arranged in XTT for them to satisfy a unicity principle by which they earn the name "equality" rather than "path".

$$\frac{\Psi, i \mid \Gamma \vdash M : A}{\Psi \mid \Gamma \vdash (\lambda i.M)(r) = M\langle r/i\rangle : A\langle r/i\rangle}$$

Function extensionality

A benefit of the cubical formulation of equality types is that the principle of function extensionality is trivially derivable in a computationally well-behaved way. Suppose that $f, g : (x : A) \to B$ and we have a family of equalities $h : (x : A) \to \mathsf{Eq}_{\_.B}(f(x), g(x))$; then, we obtain a proof that f equals g by abstraction and application:

$$\lambda i.\lambda x.h(x)(i) : \mathsf{Eq}_{\_.(x:A)\to B}(f, g)$$

In semantics of type theory, the structure of equality on a type usually mirrors the structure of the *elements* of that type in a straightforward way: for instance, a function of equations is used to equate two functions, and a pair of equations is used to equate two pairs. The benefit of the cubical approach is that this observation, at first purely empirical, is systematized by *defining* equality in every type in terms of the elements of that type in a context extended by a dimension.

Judgmental unicity of equality: boundary separation

In keeping with our desire to provide *convenient* syntax for working with extensional equality, we want proofs $P, Q : \mathsf{Eq}_{i.A}(N_0, N_1)$ of the same equation to be judgmentally equal. Rather than adding a rule to that effect, whose justification in the presence of the elimination rules for equality types would be unclear, we instead impose a more primitive *boundary separation* principle at the judgmental level: every term is completely determined by its boundary.[3]

$$\frac{\Psi \mid r\ dim \qquad \overrightarrow{\Psi, r = \varepsilon \mid \Gamma \vdash M = N : A}}{\Psi \mid \Gamma \vdash M = N : A}$$

From these rules, we can *derive* a rule that (judgmentally) equates all such $P, Q : \mathsf{Eq}_{i.A}(N_0, N_1)$.

Proof. If $P, Q : \mathsf{Eq}_{i.A}(M, N)$, then to show that $P = Q$, it suffices to show that $\lambda i.P(i) = \lambda i.Q(i)$; by the congruence rule for equality abstraction, it suffices to show that $P(i) = Q(i)$ in the extended context. But by the rules above, we may pivot on the boundary of i, and it suffices to show that $P(0) = Q(0)$ and $P(1) = Q(1)$. But these are automatic, because P and Q are both proofs of $\mathsf{Eq}_{i.A}(N_0, N_1)$, and therefore $P(\varepsilon) = Q(\varepsilon) = N_\varepsilon$. ◀

In an unpublished note from 2017, Thierry Coquand identifies a class of cubical sets equivalent to our separated types, calling them "Bishop sets" [19].

2.1 Kan operations: coercion and composition

How does one *use* a proof of equality? As we demanded in Section 1.1, we must have at least a coercion operation which, given a proof $Q : \mathsf{Eq}_{\_.\mathcal{U}_k}(A, B)$, transforms elements $M : A$ to elements of B in a coherent way.

[3] We call this principle "boundary separation" because it turns out to be exactly the fact that the collections of types and elements, when arranged into presheaves on the category of contexts, are *separated* with respect to a certain coverage on this category. We develop this perspective in Appendix B.

2.1.1 Generalized coercion

In XTT, coercion and its coherence are obtained as instances of one general operation: for any two dimensions r, r' and a *line* of types $i.C$, if M is an element of $C\langle r/i\rangle$, then $[i.C]\downarrow_{r'}^{r} M$ is an element of $C\langle r'/i\rangle$.

$$\frac{\Psi \mid r, r' \ \mathit{dim} \qquad \Psi, i \mid \Gamma \vdash C \ \mathit{type}_k \qquad \Psi \mid \Gamma \vdash M : C\langle r/i\rangle}{\Psi \mid \Gamma \vdash [i.C]\downarrow_{r'}^{r} M : C\langle r'/i\rangle}$$

In the case of a proof $Q : \mathsf{Eq}_{\_.\mathcal{U}_k}(A, B)$ of equality between types, we coerce $M : A$ to the type B using the instance $[i.Q(i)]\downarrow_1^0 M$. But how does M relate to its coercion? The principle of *coherence* demands their equality, although such an equation must relate terms of (formally) different types; this heterogeneous equality is stated in XTT using a *dependent* equality type $\mathsf{Eq}_{i.Q(i)}(M, [i.Q(i)]\downarrow_1^0 M)$. To construct an element of this equality type, we use the same coercion operator but with a different choice of r, r'; we construct this *filler* by coercing from 0 to a fresh dimension, obtaining $\lambda j.[i.Q(i)]\downarrow_j^0 M : \mathsf{Eq}_{i.Q(i)}(M, [i.Q(i)]\downarrow_1^0 M)$:

$$j.Q(j) \quad \ni \qquad M \ \xrightarrow{\ j.[i.Q(i)]\downarrow_j^0 M\ } \ [i.Q(i)]\downarrow_1^0 M$$

To see that the filler $[i.Q(i)]\downarrow_j^0 M$ has the correct boundary with respect to j, we inspect its instances under the substitutions $0/j, 1/j$. First, we observe that the right-hand side $([i.Q(i)]\downarrow_j^0 M)\langle 1/j\rangle$ is exactly $[i.Q(i)]\downarrow_1^0 M$; second, we must see that $([i.Q(i)]\downarrow_j^0 M)\langle 0/j\rangle$ is M, bringing us to an important equation that we must impose for general coercions:

$$\Psi \mid \Gamma \vdash [i.A]\downarrow_r^r M = M : A\langle r/i\rangle$$

How do coercions compute?

In order to ensure that proofs in XTT can be computed to a canonical form, we need to explain generalized coercion in each type in terms of the elements of that type. To warm up, we explain how coercion must compute in a non-dependent function type:

$$[i.A \to B]\downarrow_{r'}^{r} M = \lambda x.[i.B]\downarrow_{r'}^{r} \big(M([i.A]\downarrow_r^{r'} x)\big)$$

That is, we abstract a variable $x : A\langle r'/i\rangle$ and need to obtain an element of type $B\langle r'/i\rangle$. By *reverse* coercion, we obtain $[i.A]\downarrow_r^{r'} x : A\langle r/i\rangle$; by applying M to this, we obtain an element of type $B\langle r/i\rangle$. Finally, we coerce from r to r'. The version for dependent function types is not much harder, but requires a filler:

$$\frac{\widetilde{x} \triangleq \lambda j.[i.A]\downarrow_j^{r'} x}{[i.(x : A) \to B]\downarrow_{r'}^{r} M = \lambda x.[i.B[\widetilde{x}(i)/x]]\downarrow_{r'}^{r} M(\widetilde{x}(r))}$$

The case for dependent pair types is similar, but without the contravariance:

$$\frac{\widetilde{M_0} \triangleq \lambda j.[i.A]\downarrow_j^{r} \mathsf{fst}(M)}{[i.(x : A) \times B]\downarrow_{r'}^{r} M = \langle \widetilde{M_0}(r'), [i.B[\widetilde{M_0}(i)/x]]\downarrow_{r'}^{r} \mathsf{snd}(M)\rangle}$$

Coercions for base types (like bool) are uniformly determined by the *principle of regularity*: if A is a type which doesn't vary in the dimension i, then $[i.A]\downarrow_{r'}^{r} M$ is just M. This type-checks because $A\langle r/i\rangle = A = A\langle r'/i\rangle$; semantically, the principle of regularity is more difficult to justify in the presence of standard universes, and is not known to be compatible with principles like univalence.[4] But XTT is specifically designed to provide a theory of *equality* rather than *paths*, so we do not expect or desire to justify univalence at this level.[5]

The only difficult case is to define coercion for equality types; at first, we might try to define $[i.\mathsf{Eq}_{j.A}(N_0, N_1)] \downarrow_{r'}^{r} P$ as $\lambda j.[i.A] \downarrow_{r'}^{r} P(j)$, but this does not make type-sense: we need to see that $\big([i.A] \downarrow_{r'}^{r} P(j)\big)\langle\varepsilon/j\rangle = N_\varepsilon$, but we only obtain $\big([i.A] \downarrow_{r'}^{r} P(j)\big)\langle\varepsilon/j\rangle = [i.A] \downarrow_{r'}^{r} N_\varepsilon$, which is "off by" a coercion. Intuitively, we can solve this problem by specifying what values a coercion takes under certain substitutions; in this case, N_0 under $0/i$, and N_1 under $1/i$. We call the resulting operation *generalized composition*.

2.1.2 Generalized composition

For any dimensions r, r', s and a line of types $i.C$, if M is an element of $C\langle r/i\rangle$ and $i.N_0, i.N_1$ are lines of elements of A defined respectively on the subcubes $(s = 0), (s = 1)$ such that $N_\varepsilon\langle r/i\rangle = M$, then $[i.C] \downarrow_{r'}^{r} M\ [s \ \mathsf{with}\ 0 \hookrightarrow i.N_0 \mid 1 \hookrightarrow i.N_1]$ is an element of $C\langle r'/i\rangle$. This is called the *composite* of M with N_0, N_1 from r to r', schematically abbreviated $[i.C] \downarrow_{r'}^{r} M\ [s \ \mathsf{with}\ \overrightarrow{\varepsilon \hookrightarrow i.N_\varepsilon}]$. As with coercion, when $r = r'$, we have $[i.C] \downarrow_{r'}^{r} M\ [s \ \mathsf{with}\ \overrightarrow{\varepsilon \hookrightarrow i.N_\varepsilon}] = M$, and moreover, if $s = \varepsilon$, we have $[i.C] \downarrow_{r'}^{r} M\ [s \ \mathsf{with}\ \overrightarrow{\varepsilon \hookrightarrow i.N_\varepsilon}] = N_\varepsilon\langle r'/i\rangle$.

Returning to coercion for equality types, we now have exactly what we need:

$$[i.\mathsf{Eq}_{j.C}(N_0, N_1)] \downarrow_{r'}^{r} P = \lambda j.([i.C] \downarrow_{r'}^{r} P(j)\ [j \ \mathsf{with}\ \overrightarrow{\varepsilon \hookrightarrow \_.N_\varepsilon}])$$

Next we must explain how the generalized composition operation computes at each type; in previous works [9], we have seen that it is simpler to instead *define* generalized composition in terms of a simpler *homogeneous* version, in which one composes in a type C rather than a line of types $i.C$; we write $C \downarrow_{r'}^{r} M\ [s \ \mathsf{with}\ \overrightarrow{\varepsilon \hookrightarrow j.N_\varepsilon}]$ for this homogeneous composition, defining the generalized composition in terms of it as follows:

$$[i.C] \downarrow_{r'}^{r} M\ [s \ \mathsf{with}\ \overrightarrow{\varepsilon \hookrightarrow i.N_\varepsilon}] = C\langle r'/i\rangle \downarrow_{r'}^{r} ([i.C] \downarrow_{r'}^{r} M)\ [s \ \mathsf{with}\ \overrightarrow{\varepsilon \hookrightarrow i.[i.C] \downarrow_{r'}^{i} N_\varepsilon}]$$

Surprisingly, in XTT we do not need to build in any computation rules for homogeneous composition, because they are completely determined by judgmental boundary separation. For instance, we can derive a computation rule already for homogeneous composition in the dependent function type, by observing that the equands have the same boundary with respect to the dimension j:

$$(x : A) \to B \downarrow_{r'}^{r} M\ [j \ \mathsf{with}\ \overrightarrow{\varepsilon \hookrightarrow i.N_\varepsilon}] = \lambda x.B \downarrow_{r'}^{r} M(x)\ [j \ \mathsf{with}\ \overrightarrow{\varepsilon \hookrightarrow i.N_\varepsilon}]$$

From homogeneous composition, we obtain *symmetry and transitivity* for the equality types. Given $P : \mathsf{Eq}_{\_.A}(M, N)$, we obtain an element of type $\mathsf{Eq}_{\_.A}(N, M)$ as follows:

$$\lambda i.A \downarrow_{1}^{0} P(0)\ [i \ \mathsf{with}\ 0 \hookrightarrow j.P(j) \mid 1 \hookrightarrow \_.P(0)]$$

Furthermore, given $Q : \mathsf{Eq}_{\_.A}(N, O)$, we obtain an element of type $\mathsf{Eq}_{\_.A}(M, O)$ as follows:

$$\lambda i.A \downarrow_{1}^{0} P(i)\ [i \ \mathsf{with}\ 0 \hookrightarrow \_.P(0) \mid 1 \hookrightarrow j.Q(j)]$$

---

[4] Regularity is proved by Swan to be incompatible with univalence assuming that certain standard techniques are used [40]; however, it is still possible that there is a different way to model univalent universes with regularity. Awodey constructs a model of intensional type theory *without* universes in regular Kan cubical sets [10], using the term *normality* for what we have called regularity.

[5] Indeed, the unicity criterion of equality is also incompatible with univalence. XTT is, however, compatible with a formulation in which it is just one level of a two-level type theory, along the lines of Voevodsky's Homotopy Type System, in which the other level would have a univalent notion of path that coexists in harmony with our notion of equality [42, 9].

▶ **Example 2.1** (Identity type). It is possible to *define* Martin-Löf's identity type and its eliminator, albeit with a much stronger computation rule than is customary.

$$\mathsf{Id}_A(M,N) \triangleq \mathsf{Eq}_{\_.A}(M,N) \qquad\qquad \mathsf{refl}_A(M) \triangleq \lambda\_.M$$

$$\frac{\widetilde{P} \triangleq \lambda j.(A \downarrow_j^0 P(0)\ [i \text{ with } 0 \hookrightarrow \_.P(0) \mid 1 \hookrightarrow k.P(k)])}{\mathsf{J}_{x,y,p.C(x,y,p)}(P; x.Q(x)) \triangleq [i.C(P(0),P(i),\widetilde{P})] \downarrow_1^0 Q(P(0))}$$

2.2 Closed universes and type-case

In Section 2.1.1, we showed how to calculate coercions $[i.C] \downarrow_{r'}^{r} M$ in each type former C. In previous cubical type theories [17, 9], one could "uncover" all the things that a coercion must be equal to by reducing according to the rules which inspect the interior of the type line $i.C$; unfortunately, applying this strategy in XTT would fail to uncover certain reductions.

Specifically, given a variable $q : \mathsf{Eq}_{\_.\mathcal{U}_k}(A_0 \to B_0, A_1 \to B_1)$, the coercion $[i.q(i)] \downarrow_{r'}^{r} M$ is *not* necessarily stuck, unlike in other cubical type theories. Suppose that we can find further proofs $Q_A : \mathsf{Eq}_{\_.\mathcal{U}_k}(A_0, A_1)$ and $Q_B : \mathsf{Eq}_{\_.\mathcal{U}_k}(B_0, B_1)$; in this case, $\lambda i.Q_A(i) \to Q_B(i)$ is *also* a proof of $\mathsf{Eq}_{\_.\mathcal{U}_k}(A_0 \to B_0, A_1 \to B_1)$, so by boundary separation it must be equal to q, and therefore $[i.q(i)] \downarrow_{r'}^{r} M$ must be equal to $[i.Q_A(i) \to Q_B(i)] \downarrow_{r'}^{r} M$. But the type-directed reduction rule for coercion applies only to the latter! Generally, to see how to reduce the first coercion, it seems that we need to be able to "dream up" proofs Q_A, Q_B out of thin air, or determine that they can't exist, an impossible task.

In XTT, we cut this Gordian knot by ensuring that Q_A, Q_B *always* exist, following the approach employed in OTT. To invert the equation q into Q_A and Q_B, we add an intensional *type-case* operator to XTT, committing to a closed and inductive notion of universe by allowing pattern-matching on types [35]. It is also possible to extend XTT with open and/or univalent universes which themselves lack boundary separation, as in two-level type theories.

For illustrative purposes, consider coercion along an equality between dependent function types. Given $q : \mathsf{Eq}_{\_.\mathcal{U}_k}((x : A_0) \to B_0, (x : A_1) \to B_1)$, we define by type-case the following:

$$Q_A \triangleq \lambda i.\mathsf{case}\ q(i)\ \mathsf{of}\ [\Pi_A B \mapsto A \mid \_ \mapsto \mathsf{bool}] : \mathsf{Eq}_{\_.\mathcal{U}_k}(A_0, A_1)$$
$$Q_B \triangleq \lambda i.\mathsf{case}\ q(i)\ \mathsf{of}\ [\Pi_A B \mapsto B \mid \_ \mapsto \lambda\_.\mathsf{bool}] : \mathsf{Eq}_{i.Q_A(i) \to \mathcal{U}_k}(\lambda x.B_0, \lambda x.B_1)$$

Because of q's boundary, we are concerned only with the Π branch of the above expressions, and are free to emit a "dummy" answer in other branches. With Q_A, Q_B in hand, we note that $q(i) = (x : Q_A(i)) \to Q_B(i)(x)$ using boundary separation; therefore, we are free to calculate $[i.q(i)] \downarrow_{r'}^{r} M$ as follows:

$$\frac{\widetilde{x} \triangleq \lambda j.[i.Q_A(i)] \downarrow_j^{r'} x}{[i.q(i)] \downarrow_{r'}^{r} M = \lambda x.[i.Q_B(i)(\widetilde{x}(i))] \downarrow_{r'}^{r} M(\widetilde{x}(r))}$$

This *lazy* style of computing with proofs of equality means, in particular, that coercing along an equation cannot tell the difference between a postulated axiom and a canonical proof of equality, making XTT compatible with extension by consistent equational axioms.

3 Algebraic model theory and canonicity

We have been careful to formulate the XTT language in a *(generalized) algebraic* way, obtaining automatically a category of algebras and homomorphisms which is equipped with an initial object [14, 1]. That this initial object is isomorphic to the model of XTT obtained by constraining and quotienting its raw syntax under judgmental equality (i.e. the Lindenbaum–Tarski algebra) is an instance of Voevodsky's famous Initiality Conjecture, and we do not attempt to prove it here; we merely observe that this result has been established for several simpler type theories [38, 15].

Working within the category of XTT-algebras enables us to formulate and prove results like canonicity and normalization for the *initial* XTT-algebra in an economical manner, avoiding the usual bureaucratic overhead of reduction relations and partial equivalence relations, which were the state of the art for type-theoretic metatheory prior to Shulman [36] and Coquand [20].

Because our algebraic techniques involve defining families over *only* well-typed terms already quotiented by judgmental equality, we avoid many of the technical difficulties arising from working with the raw terms of cubical type theories, including the closure under "coherent expansion" which is critical to earlier cubical metatheories [9, 28]. Our abstract gluing–based approach therefore represents a methodological advance in metatheory for cubical type theories.

▶ **Theorem 3.1** (Canonicity). *In the initial XTT-algebra, if* $\cdot \mid \cdot \vdash M : \mathsf{bool}$*, then either* $\cdot \mid \cdot \vdash M = \mathsf{true} : \mathsf{bool}$ *or* $\cdot \mid \cdot \vdash M = \mathsf{false} : \mathsf{bool}$*.*

Following previous work [37], we employ for our semantics a variant of *categories with families* (cwf) [21] which supports a predicative hierarchy of universes à la Russell. A cwf begins with a *category of contexts* $\mathcal{C}$, and a presheaf of types $\mathsf{Ty}_{\mathcal{C}} : \widehat{\mathcal{C} \times \mathbb{L}}$. Here $\mathbb{L}$ is the category of *universe levels*, with objects the natural numbers and unique arrows $l \longrightarrow k$ if and only if $k \leq l$.[6] The fiber of the presheaf of types $\mathsf{Ty}_{\mathcal{C}} : \widehat{\mathcal{C} \times \mathbb{L}}$ at (Γ, k) is written $\mathsf{Ty}^k_{\mathcal{C}}(\Gamma)$, and contains the types in context Γ of universe level k; we require these fibers to be k-small. Reindexing implements simultaneous substitution $\gamma^* A$ and universe level shifting $\Uparrow^l_k A$. Next, we require a *dependent* presheaf of elements $\mathsf{El}_{\mathcal{C}} : \widehat{\int \mathsf{Ty}_{\mathcal{C}}}$ with the property that the functorial actions $\mathsf{El}_{\mathcal{C}}(\Gamma \vdash A) \longrightarrow \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \Uparrow^l_k A)$ are strictly identities, equating the fibers $\mathsf{El}_{\mathcal{C}}(\Gamma \vdash A)$ and $\mathsf{El}_{\mathcal{C}}(\Gamma \vdash \Uparrow^l_k A)$. The remaining data of a basic cwf is a *context comprehension*, which for every context Γ and type $A \in \mathsf{Ty}^k_{\mathcal{C}}(\Gamma)$ determines an extended context $\Gamma.A$ with a weakening substitution $\Gamma.A \xrightarrow{\mathbf{p}} \Gamma$ and a variable term $\mathbf{q} \in \mathsf{El}_{\mathcal{C}}(\Gamma.A \vdash A)$.

Next, we specify what further structure is required to make such a cwf into an XTT-algebra. To represent contexts Ψ semantically, we use the *augmented Cartesian cube category* $\square_+$, which adjoins to the Cartesian cube category $\square$ an initial object; from this, we obtain equalizers $0 = 1$ in addition to the equalizers $i = r$ which exist in $\square$. We then require a *split fibration* $\mathcal{C} \xrightarrow{\mathbf{u}} \square_+$ which implements the dependency of contexts Γ on cubes Ψ, and forces appropriate dimension restrictions to exist for contexts, types and elements. Finally, we specify algebraically the data with which such a cwf must be equipped in order to model all the connectives of XTT (see Appendix B); to distinguish the abstract (De Bruijn) syntax of the cwf from the raw syntax of XTT we use boldface, writing $\mathbf{\Pi}(A, B)$, $\mathbf{papp}(i.A, M, r)$ and $\mathbf{U}_k$ to correspond to $(x : A) \to B$, $\mathsf{app}_{i.A}(M, r)$ and $\mathcal{U}_k$ respectively, etc.

Any model of extensional type theory can be used to construct a model of XTT, so long as it is equipped with a cumulative, inductively defined hierarchy of universes closed under dependent function types, dependent pair types, extensional equality types and booleans. The interpretation of XTT into extensional models involves erasing dimensions, coercions and compositions, one instance being meaning explanations in the style of Martin-Löf [33].

6 Observe that $\mathbb{L} = \omega^{\mathsf{op}}$; reversing arrows allows us to move types from smaller universes to larger ones.

3.1 The cubical logical families construction

Any XTT-algebra $\mathcal{C}$ extends to a category $\mathcal{C}^\star$ of *proof-relevant logical predicates*, which we will call *logical families* by analogy. The proof-relevant character of the construction enables a simpler proof of canonicity than is obtained with proof-irrelevant techniques, such as partial equivalence relations. Logical families are a type-theoretic version of the *categorical gluing* construction, in which a very rich semantic category (such as sets) is cut down to include just the morphisms which track definable morphisms in $\mathcal{C}$;[7] one then uses the rich structure of the semantic category to obtain metatheoretic results about syntax (choosing $\mathcal{C}$ to be the initial model) without considering raw terms at any point in the process.

Usually, to prove canonicity one glues the initial model $\mathcal{C}$ together with **Set** along the global sections functor; this equips each context Γ with a family of sets $\Gamma^\bullet$ indexed in the *closing substitutions* for Γ. In order to prove canonicity for a cubical language like XTT, we will need a more sophisticated version of this construction, in which the global sections functor is replaced with something that determines substitutions which are closed with respect to term variables, but open with respect to dimension variables.

The split fibration $\mathcal{C} \xrightarrow{\mathfrak{u}} \square_+$ induces a functor $\square_+ \xrightarrow{\langle - \rangle} \mathcal{C}$ which takes every cube Ψ to the empty variable context over Ψ. This functor in turn induces a *nerve* construction $\mathcal{C} \xrightarrow{(\!| - |\!)} \widehat{\square}_+$, taking Γ to the cubical set $\mathcal{C}(\langle - \rangle, \Gamma)$.[8] Intuitively, this is the presheaf of substitutions which are closed with respect to term variables, but open with respect to dimension variables; when wearing $\widehat{\square}_+$-tinted glasses, these appear to be the closed substitutions.

This nerve construction extends to the presheaves of types and elements; we define the fiber of $(\!|\mathsf{Ty}_k|\!) : \widehat{\square}_+$ at Ψ to be the set $\mathsf{Ty}^k_{\mathcal{C}}(\langle\Psi\rangle)$; likewise, we define the fiber of $(\!|\mathsf{El}_k|\!) : \widehat{\int (\!|\mathsf{Ty}_k|\!)}$ at (Ψ, A) to be the set $\mathsf{El}_{\mathcal{C}}(\langle\Psi\rangle \vdash A)$. Internally to $\widehat{\square}_+$, we (abusively) write $(\!|A|\!)$ for the dependent type instance determined by $A : (\!|\mathsf{Ty}_k|\!)$.

Category of cubical logical families

Gluing $\mathcal{C}$ together with $\widehat{\square}_+$ along $(\!| - |\!)$ gives us a category of *cubical logical families* $\mathcal{C}^\star$ whose objects are pairs $\overline{\Gamma} = (\Gamma, \Gamma^\bullet)$, with $\Gamma : \mathcal{C}$ and $\Gamma^\bullet$ a *dependent cubical set* over the cubical set $(\!|\Gamma|\!)$. In other words, $\Gamma^\bullet$ is a "Kripke logical family" on the substitutions $\langle\Psi\rangle \longrightarrow \Gamma$ which commutes with dimension substitutions $\Psi' \longrightarrow \Psi$. A morphism $\overline{\Delta} \longrightarrow \overline{\Gamma}$ is a substitution $\Delta \xrightarrow{\gamma} \Gamma$ together with a proof that γ preserves the logical family: that is, an closed element $\gamma^\bullet$ of the type $\prod_{\delta : (\!|\Delta|\!)} \Delta^\bullet\delta \to \Gamma^\bullet(\gamma^*\delta)$ in the internal type theory of $\widehat{\square}_+$. We write $\overline{\gamma}$ for the pair $(\gamma, \gamma^\bullet)$. We have a fibration $\mathcal{C}^\star \xrightarrow{\pi_{\mathsf{syn}}} \mathcal{C}$ which merely projects Γ from $\overline{\Gamma} = (\Gamma, \Gamma^\bullet)$.

Glued type structure

Because we need to model *closed* universes (see Section 2.2), the standard construction of open universes in presheaf models and their gluing categories is not available to us. Therefore, we define for each $n \in \mathbb{N}$, an inductive cubical set $\mathfrak{U}^\bullet_n A : \mathcal{V}_{n+1}$ indexed over $A : (\!|\mathsf{Ty}_n|\!)$; internally to $\widehat{\square}_+$, the cubical set $\mathfrak{U}^\bullet_n A$ is the collection of *realizers* for the $\mathcal{C}$-type A. An

[7] The gluing construction is similar to realizability; the main difference is that in gluing, one considers collections of "realizers" which are *not* all drawn from a single computational domain.

[8] This construction is also called the *relative hom functor* by Fiore [22]; its use in logic originates in the study of definability for λ-calculus, characterizing the domains of discourse for Kripke logical predicates of varying arity [29]. We learned the connection to the abstract nerve construction in conversations with M. Fiore about his unpublished joint work with S. Awodey.

$$\frac{(j < n)}{\mathtt{univ}_j : \mathfrak{U}^\bullet_n \mathbf{U}_j} \qquad \frac{}{\mathtt{bool} : \mathfrak{U}^\bullet_n \mathbf{bool}}$$

$$\frac{\mathtt{A} : \mathfrak{U}^\bullet_n a \qquad \mathtt{B} : \prod_{M:(\!|A|\!)} \mathtt{A}^\circ M \to \mathfrak{U}^\bullet_n(\langle \mathbf{id}, M\rangle^* B)}{\mathtt{pi}(\mathtt{A};\mathtt{B}) : \mathfrak{U}^\bullet_n \mathbf{\Pi}(A,B) \qquad \mathtt{sg}(\mathtt{A};\mathtt{B}) : \mathfrak{U}^\bullet_n \mathbf{\Sigma}(A,B)} \qquad \frac{\mathtt{A} : \prod_{i:\mathbb{I}} \mathfrak{U}^\bullet_n A_i \qquad \overrightarrow{N^\bullet_\varepsilon : \mathtt{A}(\varepsilon)^\circ N_\varepsilon}}{\mathtt{eq}(\mathtt{A}; N^\bullet_0, N^\bullet_1) : \mathfrak{U}^\bullet_n \mathbf{Eq}(i.A_i, N_0, N_1)}$$

$$\begin{aligned}
\mathtt{univ}^\circ_n A &= \mathfrak{U}^\bullet_n A \\
\mathtt{bool}^\circ M &= (M = \mathbf{true}) + (M = \mathbf{false}) \\
\mathtt{pi}(\mathtt{A};\mathtt{B})^\circ M &= \textstyle\prod_{N:(\!|A|\!)} \prod_{N^\bullet:\mathtt{A}^\bullet N} (\mathtt{B} N N^\bullet)^\circ \mathbf{app}(A, B, M, N) \\
\mathtt{sg}(\mathtt{A};\mathtt{B})^\circ M &= \textstyle\sum_{M^\bullet_0:\mathtt{A}^\circ \mathbf{fst}(A,B,M)} (\mathtt{B}(\mathbf{fst}(A,B,M)) M^\bullet_0)^\circ \mathbf{snd}(A,B,M) \\
\mathtt{eq}(\mathtt{A}; N^\bullet_0, N^\bullet_1)^\circ M &= \left\{ M^\bullet : \textstyle\prod_{i:\mathbb{I}} \mathtt{A}(i)^\circ \mathbf{app}(A,B,M,i) \mid \overrightarrow{M^\bullet(\varepsilon) = N^\bullet_\varepsilon} \right\}
\end{aligned}$$

$$\begin{aligned}
[i.\mathtt{bool}] \downarrow^r_{r'} M^\bullet &= M^\bullet \\
[i.\mathtt{pi}(\mathtt{A};\mathtt{B})] \downarrow^r_{r'} M^\bullet &= \lambda N^\bullet.\, [i.\mathtt{B}([i.\mathtt{A}] \downarrow^{r'}_i N^\bullet)] \downarrow^r_{r'} M^\bullet\big([i.\mathtt{A}] \downarrow^{r'}_r N^\bullet\big) \\
[i.\mathtt{eq}(\mathtt{A}; N^\bullet_0, N^\bullet_1)] \downarrow^r_{r'} M^\bullet &= \lambda k.\, [i.\mathtt{A}k] \downarrow^r_{r'} M^\bullet k\ [k \text{ with } \overrightarrow{\varepsilon \hookrightarrow _.N^\bullet_\varepsilon}] \\
\mathtt{pi}(\mathtt{A};\mathtt{B}) \downarrow^r_{r'} M^\bullet\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.M'^\bullet i}] &= \lambda N^\bullet.\, \mathtt{B} N^\bullet \downarrow^r_{r'} M^\bullet N^\bullet\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.M'^\bullet i N N^\bullet}] \\
\mathtt{eq}(\mathtt{A}; N^\bullet_0, N^\bullet_1) \downarrow^r_{r'} M^\bullet\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.M'^\bullet i}] &= \lambda j.\, \mathtt{A}j \downarrow^r_{r'} M^\bullet j\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.M'^\bullet i j}] \\
&\vdots
\end{aligned}$$

Figure 2 The inductive definition of realizers $\mathfrak{U}^\bullet_n A : \mathcal{V}_{n+1}$ for types $A : (\!|\mathsf{Ty}_n|\!)$ in $\widehat{\square}_+$; we also include a fragment of the realizers for Kan operations, which are also defined by recursion on the realizers for types.

imprecise but helpful analogy is to think of a realizer $\mathtt{A} : \mathfrak{U}^\bullet_n A$ as something like the whnf of A, with the caveat that $\mathtt{A}$ is a semantic object. Simultaneously, for each $\mathtt{A} : \mathfrak{U}^\bullet_n A$, we define a cubical family $\mathtt{A}^\circ : (\!|A|\!) \to \mathcal{V}_n$ of realizers of elements of A, with each $\mathtt{A}^\circ$ being the *logical family* of the $\mathcal{C}$-type A; finally, we also define realizers for coercion and composition by recursion on the realizers for types.[9] A fragment of this definition is summarized in Figure 2.

From all this, we obtain a presheaf of types $\mathsf{Ty}_{\mathcal{C}^\star} : \widehat{\mathcal{C}^\star \times \mathbb{L}}$, taking $\mathsf{Ty}^k_{\mathcal{C}^\star}(\overline{\Gamma})$ to be the set of pairs $\overline{A} = (A, A^\bullet)$ where $A \in \mathsf{Ty}^k_{\mathcal{C}}(\Gamma)$ and $A^\bullet$ is an element of the type $\prod_{\gamma:(\!|\Gamma|\!)} \prod_{\gamma^\bullet:\Gamma^\bullet} \mathfrak{U}^\bullet_k(\gamma^* A)$ in the internal type theory of $\widehat{\square}_+$. To define the dependent presheaf of elements, we take $\mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma} \vdash \overline{A})$ to be the set of pairs $\overline{M} = (M, M^\bullet)$ where $M \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash A)$ and $M^\bullet$ is an element of the type $\prod_{\gamma:(\!|\Gamma|\!)} \prod_{\gamma^\bullet:\Gamma^\bullet} (A^\bullet \gamma\gamma^\bullet)^\circ(\gamma^* M)$ in the internal type theory of $\widehat{\square}_+$. The context comprehension $\overline{\Gamma}.\overline{A}$ is obtained by defining $\big(\overline{\Gamma}.\overline{A}\big)^\bullet\langle\gamma, M\rangle$ to be the cubical

[9] It is important to note that we do *not* use large induction-recursion in $\widehat{\square}_+$ (to our knowledge, the construction of inductive-recursive definitions has not yet been lifted to presheaf toposes); instead, we model n object universes using the meta-universe $\mathcal{V}_{n+1}$. This is an instance of *small induction-recursion*, which can be translated into indexed inductive definitions which exist in every presheaf topos [24].

set $\sum_{\gamma^\bullet:\Gamma^\bullet}(A^\bullet\gamma\gamma^\bullet)^\circ(\gamma^* M)$; it is easy to see that we obtain realizers for the weakening substitution and the variable term.

▶ **Theorem 3.2.** *$\mathcal{C}^\star$ is an XTT-algebra, and moreover, $\mathcal{C}^\star \xrightarrow{\pi_{\mathsf{syn}}} \mathcal{C}$ is a homomorphism of XTT-algebras.*

3.2 Canonicity theorem

Because $\mathcal{C}^\star$ is an XTT-algebra, we are now equipped to prove a canonicity theorem for the initial XTT-algebra $\mathcal{C}$: if M is an element of type **bool** in the empty context, then either $M = \mathbf{true}$ or $M = \mathbf{false}$.

Proof of Canonicity (Theorem 3.1). We have $M \in \mathsf{El}_{\mathcal{C}}(\cdot \vdash \mathbf{bool})$, and therefore $[\![M]\!] \in \mathsf{El}_{\mathcal{C}^\star}(\cdot \vdash \overline{\mathbf{bool}})$. From this we obtain $N : \mathsf{El}_{\mathcal{C}}(\cdot \vdash \mathbf{bool})$ where $N = \pi_{\mathsf{syn}}[\![M]\!]$, and $N^\bullet \in \mathtt{bool}^\circ_\cdot(N)$; by definition, $N^\bullet$ is either a proof that $N = \mathbf{true}$ or a proof that $N = \mathbf{false}$ (see Figure 2). Therefore, it suffices to observe that $\pi_{\mathsf{syn}}[\![M]\!] = M$; but this follows from the universal property of the initial XTT-algebra and the fact that $\mathcal{C}^\star \xrightarrow{\pi_{\mathsf{syn}}} \mathcal{C}$ is an XTT-homomorphism. ◀

References

1 Andreas Abel, Thierry Coquand, and Peter Dybjer. On the algebraic foundation of proof assistants for intuitionistic type theory. In *Functional and Logic Programming*, 2008.

2 Stuart Frazier Allen. A non-type-theoretic semantics for type-theoretic language, 1987.

3 Thorsten Altenkirch. Extensional equality in intensional type theory. In *Proceedings. 14th Symposium on Logic in Computer Science (Cat. No. PR00158)*, July 1999.

4 Thorsten Altenkirch and Conor McBride. Towards Observational Type Theory, 2006.

5 Thorsten Altenkirch, Conor McBride, and Wouter Swierstra. Observational equality, now! In *Proceedings of the 2007 Workshop on Programming Languages Meets Program Verification*, PLPV '07, 2007.

6 Carlo Angiuli, Guillaume Brunerie, Thierry Coquand, Kuen-Bang Hou (Favonia), Robert Harper, and Daniel R. Licata. Cartesian cubical type theory. December 2017.

7 Carlo Angiuli, Evan Cavallo, Kuen-Bang Hou (Favonia), Robert Harper, Anders Mörtberg, and Jonathan Sterling. `redtt`: implementing Cartesian cubical type theory.

8 Carlo Angiuli, Evan Cavallo, Kuen-Bang Hou (Favonia), Robert Harper, and Jonathan Sterling. The RedPRL Proof Assistant (Invited Paper). In *Proceedings of the 13th International Workshop on Logical Frameworks and Meta-Languages: Theory and Practice, LFMTP@FSCD 2018, Oxford, UK, 7th July 2018.*, 2018.

9 Carlo Angiuli, Kuen-Bang Hou (Favonia), and Robert Harper. Cartesian Cubical Computational Type Theory: Constructive Reasoning with Paths and Equalities. In *27th EACSL Annual Conference on Computer Science Logic (CSL 2018)*, volume 119 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 2018.

10 Steve Awodey. A cubical model of homotopy type theory. *Annals of Pure and Applied Logic*, 169(12), 2018.

11 Steve Awodey. Natural models of homotopy type theory. *Mathematical Structures in Computer Science*, 28(2), 2018.

12 Andrej Bauer, Gaëtan Gilbert, Philipp Haselwarter, Matija Pretnar, and Christopher A. Stone. Design and implementation of the Andromeda proof assistant. 2016.

13 Errett Bishop. *Foundations of Constructive Analysis.* 1967.

14 John Cartmell. Generalised algebraic theories and contextual categories. *Annals of Pure and Applied Logic*, 32, 1986.

15 Simon Castellan, Pierre Clairambault, and Peter Dybjer. Undecidability of equality in the free locally cartesian closed category (extended version). *Logical Methods in Computer Science*, 13(4), 2017.

16 James Chapman, Fredrik Nordvall Forsberg, and Conor McBride. The Box of Delights (Cubical Observational Type Theory). January 2018.

17 Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. Cubical Type Theory: a constructive interpretation of the univalence axiom. *IfCoLog Journal of Logics and their Applications*, 4(10), November 2017.

18 R. L. Constable, S. F. Allen, H. M. Bromley, W. R. Cleaveland, J. F. Cremer, R. W. Harper, D. J. Howe, T. B. Knoblock, N. P. Mendler, P. Panangaden, J. T. Sasaki, and S. F. Smith. *Implementing Mathematics with the Nuprl Proof Development System*. 1986.

19 Thierry Coquand. Universe of Bishop sets, February 2017.

20 Thierry Coquand. Canonicity and normalization for Dependent Type Theory, October 2018.

21 Peter Dybjer. Internal type theory. In *Types for Proofs and Programs: International Workshop, TYPES '95 Torino, Italy, June 5–8, 1995 Selected Papers*. 1996.

22 Marcelo Fiore. Semantic analysis of normalisation by evaluation for typed lambda calculus. In *Proceedings of the 4th ACM SIGPLAN International Conference on Principles and Practice of Declarative Programming*, PPDP '02, 2002.

23 Marcelo Fiore. Discrete generalised polynomial functors, 2012.

24 Peter Hancock, Conor McBride, Neil Ghani, Lorenzo Malatesta, and Thorsten Altenkirch. Small induction recursion. In *Typed Lambda Calculi and Applications: 11th International Conference, TLCA 2013, Eindhoven, The Netherlands, June 26-28, 2013. Proceedings*, 2013.

25 Martin Hofmann. Extensional concepts in intensional type theory, January 1995.

26 Martin Hofmann and Thomas Streicher. Lifting Grothendieck universes. 1997.

27 Martin Hofmann and Thomas Streicher. The groupoid interpretation of type theory. In *Twenty-five years of constructive type theory (Venice, 1995)*, volume 36 of *Oxford Logic Guides*. 1998.

28 Simon Huber. Canonicity for cubical type theory. *Journal of Automated Reasoning*, June 2018.

29 Achim Jung and Jerzy Tiuryn. A new characterization of lambda definability. In *Typed Lambda Calculi and Applications*, 1993.

30 Chris Kapulkin and Peter LeFanu Lumsdaine. The simplicial model of Univalent Foundations (after Voevodsky), June 2016.

31 Alexei Kopylov. Type theoretical foundations for data structures, classes and objects, 2004.

32 Per Martin-Löf. An intuitionistic theory of types: Predicative part. In *Logic Colloquium '73*, volume 80 of *Studies in Logic and the Foundations of Mathematics*. 1975.

33 Per Martin-Löf. *Intuitionistic type theory*, volume 1 of *Studies in Proof Theory*. 1984.

34 Conor McBride. Dependently typed functional programs and their proofs, 1999.

35 Bengt Nordström, Kent Peterson, and Jan M. Smith. *Programming in Martin-Lof's Type Theory*, volume 7 of *International Series of Monographs on Computer Science*. 1990.

36 Michael Shulman. Univalence for inverse diagrams and homotopy canonicity. *Mathematical Structures in Computer Science*, 25(5), 2015.

37 Jonathan Sterling. Algebraic type theory and universe hierarchies. December 2018.

38 Thomas Streicher. *Semantics of Type Theory: Correctness, Completeness, and Independence Results*. 1991.

39 Thomas Streicher. Investigations into intensional type theory. Habilitationsschrift, Universität München, 1994.

40 Andrew Swan. Separating path and identity types in presheaf models of univalent type theory. 2018.

41 The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. 2013.

42 Vladimir Voevodsky. A simple type system with two identity types. February 2013.

A The XTT language

Annotated syntax

The raw syntax of XTT includes typing annotations on function application $\mathsf{app}_{x:A.B}(M, N)$ and pair projections $\mathsf{fst}_{x:A.B}(M)$ and $\mathsf{snd}_{x:A.B}(M)$, in order to ensure that the raw syntax could (in theory) be organized into an *initial* model of XTT, in the sense of Appendix B. A version of the syntax with fewer annotations would be justified by a normalization result for XTT, which we do *not* establish here.

Because these annotations can visually obscure the meaning of a term, we adopt the notational convention that when a term is already known to be well-typed, we omit the annotation and write $M(N)$ for $\mathsf{app}_{x:A.B}(M, N)$, and likewise $\mathsf{fst}(M)$ for $\mathsf{fst}_{x:A.B}(M)$, etc.

Heterogeneous composition

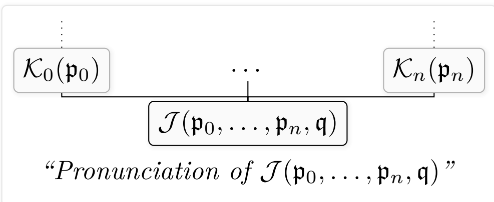
Following previous work [9], we take coercion and homogeneous composition as primitive operations, and define heterogeneous composition in terms of it:

$$[i.A] \downarrow^{r}_{r'} M\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.N_\varepsilon}] \triangleq A\langle r'/j\rangle \downarrow^{r}_{r'} ([j.A] \downarrow^{r}_{r'} M)\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.[j.A] \downarrow^{j}_{r'} N_\varepsilon}]$$

Other versions of cubical type theory, such as De Morgan cubical type theory [17], take heterogeneous composition as primitive and derive both coercion and homogeneous composition as a special case. In our setting, it is especially advantageous to take coercion and homogeneous composition as a primitive, because in XTT it is only necessary to provide β-rules for coercion; in Appendix A.3, we observe that all the β-rules for homogeneous composition are in fact already derivable, by exploiting the path unicity rule in Appendix A.2.4.

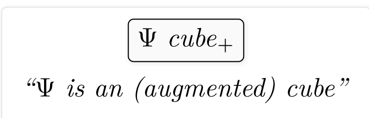
▶ **Convention A.1** (Presupposition)**.** The XTT language involves many forms of judgment, each of is defined conditionally on a *presupposition*; in type theoretic formal systems, a judgment expresses the well-formedness of a raw term (the "subject") relative to some parameters. The parameters themselves are rarely raw terms, but rather terms that are already known to be well-formed according to certain judgments (called "presuppositions").

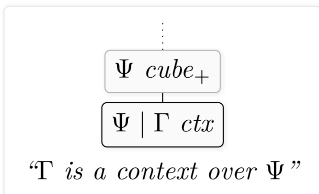
We indicate this situation schematically for a form of judgment $\mathcal{J}$ in the following way, where $\overrightarrow{\mathfrak{p}_i}$ are parameters and $\mathfrak{q}$ is a subject:



A.1 The judgments of XTT

The judgments of XTT are summarized below using Convention A.1.





$\Psi\ cube_+$

$\Psi \mid r\ dim$

"r *is a dimension over* Ψ"

$\Psi \mid r\ dim$ $\quad$ $\Psi \mid r'\ dim$

$\Psi \mid r = r'\ dim$

"r *and* r' *are equal dimensions in* Ψ"

$\Psi \mid \Gamma\ ctx$

$\Psi \mid \Gamma \vdash A\ type_k$

"A *is a type at level* k *in context* Γ"

$\Psi \mid \Gamma \vdash A\ type_k$ $\quad$ $\Psi \mid \Gamma \vdash B\ type_k$

$\Psi \mid \Gamma \vdash A = B\ type_k$

"A *and* B *are equal types at level* k"

$\Psi \mid \Gamma \vdash A\ type_k$

$\Psi \mid \Gamma \vdash M : A$

"M *is a term of type* A"

$\Psi \mid \Gamma \vdash M : A$ $\quad$ $\Psi \mid \Gamma \vdash N : A$

$\Psi \mid \Gamma \vdash M = N : A$

"M *and* N *are equal terms of type* A"

$\overrightarrow{\Psi, \xi \mid \Gamma \vdash B\ type_k}$

$\Psi \mid \Gamma \vdash A\ type_k\ [\overrightarrow{\xi \hookrightarrow B}]$

"A *is a type at level* k *which matches each* B *at* ξ"

$\Psi \mid \Gamma \vdash A\ type_k$ $\quad$ $\overrightarrow{\Psi, \xi \mid \Gamma \vdash N : A}$

$\Psi \mid \Gamma \vdash M : A\ [\overrightarrow{\xi \hookrightarrow N}]$

"M *is an element of* A *which matches each* N *at* ξ"

A.2 The rules of XTT

In the following sections, we summarize the rules of XTT; we systematically omit all congruence rules for formal equality, because these can be mechanically obtained from the typing rules.

A.2.1 Cubes

$$\text{EMP}\ \frac{}{\cdot\ cube_+ \cdot} \qquad \text{SNOC/DIM}\ \frac{\Psi\ cube_+}{\Psi, i\ cube_+} \qquad \text{SNOC/CONSTR}\ \frac{\Psi\ cube_+ \qquad \Psi \mid \Psi\ dim r, r'}{\Psi, r = r'\ cube_+}$$

A.2.2 Contexts

$$\textsc{Emp}\ \frac{}{\Psi \mid \cdot\ \mathit{ctx}} \qquad \textsc{Snoc}\ \frac{\Psi \mid \Gamma\ \mathit{ctx} \qquad \Psi \mid \Gamma \vdash A\ \mathit{type}_k}{\Psi \mid \Gamma, x : A\ \mathit{ctx}}$$

A.2.3 Dimensions

$$\textsc{Constant}\ \frac{}{\Psi \mid \varepsilon\ \mathit{dim}} \qquad \textsc{Variable}\ \frac{}{\Psi, i \mid i\ \mathit{dim}} \qquad \textsc{Reflexivity}\ \frac{}{\Psi \mid r = r\ \mathit{dim}} \qquad \textsc{Symmetry}\ \frac{\Psi \mid r = r'\ \mathit{dim}}{\Psi \mid r' = r\ \mathit{dim}}$$

$$\textsc{Transitivity}\ \frac{\Psi \mid r_0 = r_1\ \mathit{dim} \qquad \Psi \mid r_1 = r_2\ \mathit{dim}}{\Psi \mid r_0 = r_2\ \mathit{dim}} \qquad \frac{\Psi \ni r = r'}{\Psi \mid r = r'\ \mathit{dim}}$$

A.2.4 Structural

$$\textsc{Variable}\ \frac{\Gamma \ni x : A}{\Psi \mid \Gamma \vdash x : A} \qquad \textsc{False Constraint}\ \frac{\Psi \mid 0 = 1\ \mathit{dim}}{\Psi \mid \Gamma \vdash \mathcal{J}} \qquad \textsc{Conversion}\ \frac{\Psi \mid \Gamma \vdash A_0 = A_1\ \mathit{type}_k \qquad \Psi \mid \Gamma \vdash M : A_0}{\Psi \mid \Gamma \vdash M : A_1}$$

$$\textsc{Boundary Separation (Types)}\ \frac{\Psi \mid r\ \mathit{dim} \qquad \overrightarrow{\Psi, r = \varepsilon \mid \Gamma \vdash A = B\ \mathit{type}_k}}{\Psi \mid \Gamma \vdash A = B\ \mathit{type}_k} \qquad \textsc{Boundary Separation (Terms)}\ \frac{\Psi \mid r\ \mathit{dim} \qquad \overrightarrow{\Psi, r = \varepsilon \mid \Gamma \vdash M = N : A}}{\Psi \mid \Gamma \vdash M = N : A}$$

The following rules are admissible:

$$\textsc{Constraint Cut}\ \frac{\Psi \mid r = r'\ \mathit{dim} \qquad \Psi \mid \Gamma \vdash \mathcal{J}}{\Psi, r = r' \mid \Gamma \vdash \mathcal{J}} \qquad \textsc{Constraint Weakening}\ \frac{\Psi \mid \Gamma \vdash \mathcal{J}}{\Psi, \xi \mid \Gamma \vdash \mathcal{J}}$$

A.2.5 Coercion

$$\textsc{Coercion}\ \frac{\Psi \mid r, r'\ \mathit{dim} \qquad \Psi, i \mid \Gamma \vdash A\ \mathit{type}_k \qquad \Psi \mid \Gamma \vdash M : A\langle r/i \rangle}{\Psi \mid \Gamma \vdash [i.A] \downarrow^{r}_{r'} M : A\langle r'/i \rangle}$$

$$\textsc{Coercion Boundary}\ \frac{}{\Psi \mid \Gamma \vdash [i.A] \downarrow^{r}_{r} M = M : A\langle r/i \rangle} \qquad \textsc{Coercion Regularity}\ \frac{\Psi, j, k \mid \Gamma \vdash A\langle j/i \rangle = A\langle k/i \rangle\ \mathit{type}_k}{\Psi \mid \Gamma \vdash [i.A] \downarrow^{r}_{r'} M = M : A\langle r'/i \rangle}$$

A.2.6 Composition

$$\textsc{Composition}\ \frac{\Psi \mid r, r', s\ \mathit{dim} \qquad \Psi \mid \Gamma \vdash M : A \qquad \overrightarrow{\Psi, j, s = \varepsilon \mid \Gamma \vdash N_\varepsilon : A\ [j = r \hookrightarrow M]}}{\Psi \mid \Gamma \vdash A \downarrow^{r}_{r'} M\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.N_\varepsilon}] : A}$$

$$\textsc{Composition Boundary}\ \frac{}{\begin{array}{l}\Psi \mid \Gamma \vdash A \downarrow^{r}_{r} M\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.N_\varepsilon}] = M : A \\ \Psi \mid \Gamma \vdash A \downarrow^{r}_{r'} M\ [\varepsilon \text{ with } \overrightarrow{\varepsilon' \hookrightarrow j.N_{\varepsilon'}}] = N_\varepsilon\langle r'/j \rangle : A\end{array}}$$

A.2.7 Level restrictions

LIFT FORMATION
$$\frac{\Psi \mid \Gamma \vdash A\ \mathit{type}_k \qquad k \leq l}{\Psi \mid \Gamma \vdash \Uparrow^l_k A\ \mathit{type}_l}$$

LIFT ELEMENT
$$\frac{\Psi \mid \Gamma \vdash M : A}{\Psi \mid \Gamma \vdash M : \Uparrow^l_k A}$$

LIFT HYPOTHESIS
$$\frac{\Psi \mid \Gamma, x : A \vdash \mathcal{J}}{\Psi \mid \Gamma, x : \Uparrow^l_k A \vdash \mathcal{J}}$$

LIFT FUNCTORIALITY
$$\frac{}{\begin{array}{l}\Psi \mid \Gamma \vdash \Uparrow^k_k A = A\ \mathit{type}_k \\ \Psi \mid \Gamma \vdash \Uparrow^m_l \Uparrow^l_k A = \Uparrow^m_k A\ \mathit{type}_k\end{array}}$$

LIFT COERCION
$$\frac{}{\Psi \mid \Gamma \vdash [i.\Uparrow^l_k A] \downarrow^r_{r'} M = [i.A] \downarrow^r_{r'} M : \Uparrow^l_k A\langle r'/i\rangle}$$

A.2.8 Dependent pair types

PAIR FORMATION
$$\frac{\Psi \mid \Gamma \vdash A\ \mathit{type}_k \qquad \Psi \mid \Gamma, x : A \vdash B\ \mathit{type}_k}{\Psi \mid \Gamma \vdash (x : A) \times B\ \mathit{type}_k}$$

PAIR LIFTING
$$\frac{}{\Psi \mid \Gamma \vdash \Uparrow^l_k (x : A) \times B = (x : \Uparrow^l_k A) \times \Uparrow^l_k B\ \mathit{type}_l}$$

PAIR INTRODUCTION
$$\frac{\Psi \mid \Gamma \vdash M : A \qquad \Psi \mid \Gamma \vdash N : B[M/x]}{\Psi \mid \Gamma \vdash \langle M, N\rangle : (x : A) \times B}$$

PAIR ELIMINATION
$$\frac{\Psi \mid \Gamma \vdash M : (x : A) \times B}{\begin{array}{l}\Psi \mid \Gamma \vdash \mathsf{fst}_{x:A.B}(M) : A \\ \Psi \mid \Gamma \vdash \mathsf{snd}_{x:A.B}(M) : B[\mathsf{fst}(M)/x]\end{array}}$$

PAIR COMPUTATION
$$\frac{}{\begin{array}{l}\Psi \mid \Gamma \vdash \mathsf{fst}(\langle M, N\rangle) = M : A \\ \Psi \mid \Gamma \vdash \mathsf{snd}(\langle M, N\rangle) = N : B\end{array}}$$

PAIR COERCION COMPUTATION (1)
$$\frac{}{\Psi \mid \Gamma \vdash \mathsf{fst}([i.(x : A) \times B] \downarrow^r_{r'} M) = [i.A] \downarrow^r_{r'} \mathsf{fst}(M) : A[r'/i]}$$

PAIR COERCION COMPUTATION (2)
$$\frac{\Psi \mid \Gamma \vdash H \triangleq [i.B[[i.A] \downarrow^r_i \mathsf{fst}(M)/x]] \downarrow^r_{r'} \mathsf{snd}(M)}{\Psi \mid \Gamma \vdash \mathsf{snd}([i.(x : A) \times B] \downarrow^r_{r'} M) = H : B\langle r'/i\rangle[[i.A] \downarrow^r_{r'} \mathsf{fst}(M)/x]}$$

PAIR UNICITY
$$\frac{}{\Psi \mid \Gamma \vdash M = \langle \mathsf{fst}(M), \mathsf{snd}(M)\rangle : (x : A) \times B}$$

A.2.9 Dependent function types

FUNCTION FORMATION
$$\frac{\Psi \mid \Gamma \vdash A\ \mathit{type}_k \qquad \Psi \mid \Gamma, x : A \vdash B\ \mathit{type}_k}{\Psi \mid \Gamma \vdash (x : A) \to B\ \mathit{type}_k}$$

FUNCTION LIFTING
$$\frac{}{\Psi \mid \Gamma \vdash \Uparrow^l_k (x : A) \to B = (x : \Uparrow^l_k A) \to \Uparrow^l_k B\ \mathit{type}_l}$$

FUNCTION INTRODUCTION
$$\frac{\Psi \mid \Gamma, x : A \vdash M : B}{\Psi \mid \Gamma \vdash \lambda x.M : (x : A) \to B}$$

FUNCTION ELIMINATION
$$\frac{\Psi \mid \Gamma \vdash M : (x : A) \to B \qquad \Psi \mid \Gamma \vdash N : A}{\Psi \mid \Gamma \vdash \mathsf{app}_{x:A.B}(M, N) : B[N/x]}$$

FUNCTION COMPUTATION
$$\frac{\Psi \mid \Gamma, x : A \vdash M : B}{\Psi \mid \Gamma \vdash (\lambda x.M)(N) = M[N/x] : B[N/x]}$$

FUNCTION COERCION COMPUTATION

$$\frac{\Psi, i \mid \Gamma \vdash \widetilde{N}[i] \triangleq [i.A] \downarrow_i^{r'} N}{\Psi \mid \Gamma \vdash ([i.(x : A) \to B] \downarrow_{r'}^{r} M)(N) = [i.B[\widetilde{N}[i]/x]] \downarrow_{r'}^{r} M(\widetilde{N}[r]) : C}$$

FUNCTION UNICITY

$$\frac{}{\Psi \mid \Gamma \vdash M = \lambda x.M(x) : (x : A) \to B}$$

A.2.10 Dependent equality types

EQUALITY FORMATION

$$\frac{\Psi, i \mid \Gamma \vdash A \ \mathit{type}_k \qquad \overrightarrow{\Psi, i = \varepsilon \mid \Gamma \vdash N_\varepsilon : A}}{\Psi \mid \Gamma \vdash \mathsf{Eq}_{i.A}(N_0, N_1) \ \mathit{type}_k}$$

EQUALITY LIFTING

$$\frac{}{\Psi \mid \Gamma \vdash \Uparrow_k^l \mathsf{Eq}_{i.A}(N_0, N_1) = \mathsf{Eq}_{i.\Uparrow_k^l A}(N_0, N_1) \ \mathit{type}_l}$$

EQUALITY INTRODUCTION

$$\frac{\Psi, i \mid \Gamma \vdash M : A \ [\overrightarrow{i = \varepsilon \hookrightarrow N_\varepsilon}]}{\Psi \mid \Gamma \vdash \lambda i.M : \mathsf{Eq}_{i.A}(N_0, N_1)}$$

EQUALITY ELIMINATION

$$\frac{\Psi \mid r \ \mathit{dim} \qquad \Psi \mid \Gamma \vdash M : \mathsf{Eq}_{i.A}(N_0, N_1)}{\Psi \mid \Gamma \vdash M(r) : A\langle r/i\rangle}$$

EQUALITY BOUNDARY

$$\frac{\Psi \mid \Gamma \vdash M : \mathsf{Eq}_{i.A}(N_0, N_1)}{\Psi \mid \Gamma \vdash M(\varepsilon) = N_\varepsilon : A\langle \varepsilon/i\rangle}$$

EQUALITY COMPUTATION

$$\frac{\Psi, i \mid \Gamma \vdash M : A}{\Psi \mid \Gamma \vdash (\lambda i.M)(r) = M\langle r/i\rangle : A\langle r/i\rangle}$$

EQUALITY COERCION COMPUTATION

$$\frac{}{\Psi \mid \Gamma \vdash ([j.\mathsf{Eq}_{i.A}(N_0, N_1)] \downarrow_{r'}^{r} P)(s) = [j.A\langle s/i\rangle] \downarrow_{r'}^{r} P(s) \ [s \ \mathsf{with} \ \overrightarrow{\varepsilon \hookrightarrow j.N_\varepsilon}] : A\langle r', s/j, i\rangle}$$

EQUALITY UNICITY

$$\frac{}{\Psi \mid \Gamma \vdash M = \lambda i.M(i) : \mathsf{Eq}_{i.A}(N_0, N_1)}$$

A.2.11 Booleans

BOOLEAN FORMATION

$$\frac{}{\Psi \mid \Gamma \vdash \mathsf{bool} \ \mathit{type}_k}$$

BOOLEAN LIFTING

$$\frac{}{\Psi \mid \Gamma \vdash \Uparrow_k^l \mathsf{bool} = \mathsf{bool} \ \mathit{type}_l}$$

BOOLEAN INTRODUCTION

$$\frac{}{\begin{array}{c}\Psi \mid \Gamma \vdash \mathsf{true} : \mathsf{bool} \\ \Psi \mid \Gamma \vdash \mathsf{false} : \mathsf{bool}\end{array}}$$

BOOLEAN ELIMINATION

$$\frac{\begin{array}{l}\Psi \mid \Gamma, x : \mathsf{bool} \vdash C \ \mathit{type}_k \\ \Psi \mid \Gamma \vdash M : \mathsf{bool} \qquad \Psi \mid \Gamma \vdash N_0 : C[\mathsf{true}/x] \qquad \Psi \mid \Gamma \vdash N_1 : C[\mathsf{false}/x]\end{array}}{\Psi \mid \Gamma \vdash \mathsf{if}_{x.C}(M; N_0, N_1) : C[M/x]}$$

BOOLEAN ELIMINATION LIFTING

$$\frac{}{\Psi \mid \Gamma \vdash \mathsf{if}_{x.\Uparrow_k^l C}(M; N_0, N_1) = \mathsf{if}_{x.C}(M; N_0, N_1) : \Uparrow_k^l C[M/x]}$$

BOOLEAN COMPUTATION

$$\frac{}{\begin{array}{c}\Psi \mid \Gamma \vdash \mathsf{if}_{x.C}(\mathsf{true}; N_0, N_1) = N_0 : C[\mathsf{true}/x] \\ \Psi \mid \Gamma \vdash \mathsf{if}_{x.C}(\mathsf{false}; N_0, N_1) = N_1 : C[\mathsf{false}/x]\end{array}}$$

A.2.12 Universes

UNIVERSE FORMATION
$$\frac{(k < l)}{\Psi \mid \Gamma \vdash \mathcal{U}_k \ \mathit{type}_l}$$

UNIVERSE LIFTING
$$\frac{}{\Psi \mid \Gamma \vdash \Uparrow_l^m \mathcal{U}_k = \mathcal{U}_k \ \mathit{type}_m}$$

UNIVERSE ELEMENTS
$$\frac{\Psi \mid \Gamma \vdash A \ \mathit{type}_k}{\Psi \mid \Gamma \vdash A : \mathcal{U}_k}$$

UNIVERSE EQUALITY
$$\frac{\Psi \mid \Gamma \vdash A_0 = A_1 \ \mathit{type}_k}{\Psi \mid \Gamma \vdash A_0 = A_1 : \mathcal{U}_k}$$

TYPE-CASE
$$\frac{\begin{array}{l}\Psi \mid \Gamma \vdash C \ \mathit{type}_l \\ \Psi \mid \Gamma, x : \mathcal{U}_k, y : x \to \mathcal{U}_k \vdash M_\Pi : C \\ \Psi \mid \Gamma, x : \mathcal{U}_k, y : x \to \mathcal{U}_k \vdash M_\Sigma : C \\ \Psi \mid \Gamma, x_0 : \mathcal{U}_k, x_1 : \mathcal{U}_k, x^= : \mathsf{Eq}_{i.\mathcal{U}_k}(x_0, x_1), y_0 : x_0, y_1 : x_1 \vdash M_{\mathsf{Eq}} : C \\ \Psi \mid \Gamma \vdash M_{\mathsf{bool}} : C \\ \Psi \mid \Gamma \vdash M_{\mathcal{U}} : C\end{array}}{\Psi \mid \Gamma \vdash \mathsf{case}\ X\ \mathsf{of}\ [\Pi_x y \mapsto M_\Pi \mid \Sigma_x y \mapsto M_\Sigma \mid \mathsf{Eq}_{x_0,x_1,x^=}(y_0, y_1) \mapsto M_{\mathsf{Eq}} \mid \mathsf{bool} \mapsto M_{\mathsf{bool}} \mid \mathcal{U} \mapsto M_{\mathcal{U}}] : C}$$

TYPE-CASE COMPUTATION
$$\frac{}{\begin{array}{l}\Psi \mid \Gamma \vdash \mathsf{case}\ ((z : A) \to B)\ \mathsf{of}\ [\Pi_x y \mapsto M \mid \ldots] = M[A, \lambda z.B/x, y] : C \\ \Psi \mid \Gamma \vdash \mathsf{case}\ ((z : A) \times B)\ \mathsf{of}\ [\ldots \mid \Sigma_x y \mapsto M \mid \ldots] = M[A, \lambda z.B/x, y] : C \\ \Psi \mid \Gamma \vdash \mathsf{case}\ \mathsf{bool}\ \mathsf{of}\ [\ldots \mid \mathsf{bool} \mapsto M \mid \ldots] = M : C \\ \Psi \mid \Gamma \vdash \mathsf{case}\ \mathcal{U}_{k'}\ \mathsf{of}\ [\ldots \mid \mathcal{U} \mapsto M] = M : C\end{array}}$$

$$\frac{\Psi \mid \Gamma \vdash H \triangleq M[A\langle 0/i\rangle, A\langle 1/i\rangle, \lambda i.A, N_0, N_1/x_0, x_1, x^=, y_0, y_1]}{\Psi \mid \Gamma \vdash \mathsf{case}\ (\mathsf{Eq}_{i.A}(N_0, N_1))\ \mathsf{of}\ [\ldots \mid \mathsf{Eq}_{x_0,x_1,x^=}(y_0, y_1) \mapsto M \mid \ldots] = H : C}$$

A.2.13 Boundary matching

TYPE BOUNDARY
$$\frac{\Psi \mid \Gamma \vdash M : A \qquad \overrightarrow{\Psi, \xi \mid \Gamma \vdash M = N : A}}{\Psi \mid \Gamma \vdash M : A\ [\overrightarrow{\xi \hookrightarrow N}]}$$

TERM BOUNDARY
$$\frac{\Psi \mid \Gamma \vdash A \ \mathit{type}_k \qquad \overrightarrow{\Psi, \xi \mid \Gamma \vdash A = B \ \mathit{type}_k}}{\Psi \mid \Gamma \vdash A \ \mathit{type}_k\ [\overrightarrow{\xi \hookrightarrow B}]}$$

A.3 Derivable Rules

Numerous additional rules about compositions are *derivable* by exploiting boundary separation. In previous presentations of cubical type theory (which did not enjoy the unicity of equality proofs), it was necessary to include β-rules for compositions explicitly.

COMPOSITION REGULARITY
$$\frac{\overrightarrow{\Psi, j_0, j_1, i = \varepsilon \mid \Gamma \vdash N_\varepsilon\langle j_0/j\rangle = N_\varepsilon\langle j_1/j\rangle : A}}{\Psi \mid \Gamma \vdash A \downarrow_{r'}^{r} M\ [i\ \mathsf{with}\ \overrightarrow{\varepsilon \hookrightarrow j.N_\varepsilon}] = M : A}$$

HETEROGENEOUS COMPOSITION
$$\frac{\Psi \mid r, r', s \ \mathit{dim} \qquad \Psi \mid \Gamma \vdash M : A\langle r/j\rangle \qquad \overrightarrow{\Psi, j, s = \varepsilon \mid \Gamma \vdash N_\varepsilon : A\ [j = r \hookrightarrow M]}}{\Psi \mid \Gamma \vdash [j.A] \downarrow_{r'}^{r} M\ [s\ \mathsf{with}\ \overrightarrow{\varepsilon \hookrightarrow j.N_\varepsilon}] : A\langle r'/j\rangle}$$

HETEROGENEOUS COMPOSITION BOUNDARY

$$\frac{}{\begin{array}{c}\Psi \mid \Gamma \vdash [j.A] \downarrow^r_r M \ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.N_\varepsilon}] = M : A\langle r/j\rangle \\ \Psi \mid \Gamma \vdash [j.A] \downarrow^r_{r'} M \ [\varepsilon \text{ with } \overrightarrow{\varepsilon' \hookrightarrow j.N_{\varepsilon'}}] = N_\varepsilon\langle r'/j\rangle : A\langle r'/j\rangle\end{array}}$$

LIFT COMPOSITION

$$\frac{}{\Psi \mid \Gamma \vdash \Uparrow^l_k A \downarrow^r_{r'} M \ [i \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.N_\varepsilon}] = A \downarrow^r_{r'} M \ [i \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.N_\varepsilon}] : \Uparrow^l_k A}$$

LIFT TYPE COMPOSITION

$$\frac{}{\Psi \mid \Gamma \vdash \mathcal{U}_l \downarrow^r_{r'} \Uparrow^l_k A \ [i \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.\Uparrow^l_k B_\varepsilon}] = \Uparrow^l_k \mathcal{U}_k \downarrow^r_{r'} A \ [i \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.B_\varepsilon}] \ \mathit{type}_k}$$

PAIR COMPOSITION COMPUTATION (1)

$$\frac{\Psi \mid \Gamma \vdash H \triangleq A \downarrow^r_{r'} \mathsf{fst}(M) \ [i \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.\mathsf{fst}(N_\varepsilon)}]}{\Psi \mid \Gamma \vdash \mathsf{fst}((x : A) \times B \downarrow^r_{r'} M \ [i \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.N_\varepsilon}]) = H : A}$$

PAIR COMPOSITION COMPUTATION (2)

$$\frac{\begin{array}{c}\Psi, k \mid \Gamma \vdash \widetilde{M_1}[k] \triangleq A \downarrow^r_k \mathsf{fst}(M) \ [i \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.\mathsf{fst}(N_\varepsilon)}] \\ \Psi \mid \Gamma \vdash H \triangleq [k.B[\widetilde{M_1}[k]/x]] \downarrow^r_{r'} \mathsf{snd}(M) \ [i \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.\mathsf{snd}(N_\varepsilon)}]\end{array}}{\Psi \mid \Gamma \vdash \mathsf{snd}((x : A) \times B \downarrow^r_{r'} M \ [i \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.N_\varepsilon}]) = H : B[\widetilde{M_1}[r']/x]}$$

PAIR TYPE COMPOSITION

$$\frac{\begin{array}{l}\Psi, k \mid \Gamma \vdash \widetilde{A}[k] \triangleq \mathcal{U}_k \downarrow^r_k A \ [i \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.A_\varepsilon}] \\ \Psi, j \mid \Gamma, x : \widetilde{A}[r'] \vdash \widetilde{x}[j] \triangleq [k.\widetilde{A}[k]] \downarrow^{r'}_j x \\ \Psi \mid \Gamma, x : \widetilde{A}[r'] \vdash \widetilde{B} \triangleq \mathcal{U}_k \downarrow^r_{r'} B[\widetilde{x}[r]/x] \ [i \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.B_\varepsilon[\widetilde{x}[j]/x]}]\end{array}}{\Psi \mid \Gamma \vdash \mathcal{U}_k \downarrow^r_{r'} ((x : A) \times B) \ [i \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.(x : A_\varepsilon) \times B_\varepsilon}] = (x : \widetilde{A}[r']) \times \widetilde{B} \ \mathit{type}_l}$$

FUNCTION COMPOSITION COMPUTATION

$$\frac{\Psi \mid \Gamma \vdash H \triangleq B[N/x] \downarrow^r_{r'} M(N) \ [i \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.M_\varepsilon(N)}]}{\Psi \mid \Gamma \vdash ((x : A) \to B \downarrow^r_{r'} M \ [i \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.M_\varepsilon}])(N) = H : (x : A) \to B}$$

FUNCTION TYPE COMPOSITION

$$\frac{\begin{array}{l}\Psi, k \mid \Gamma \vdash \widetilde{A}[k] \triangleq \mathcal{U}_k \downarrow^r_k A \ [i \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.A_\varepsilon}] \\ \Psi, j \mid \Gamma, x : \widetilde{A}[r'] \vdash \widetilde{x}[j] \triangleq [k.\widetilde{A}[k]] \downarrow^{r'}_j x \\ \Psi \mid \Gamma, x : \widetilde{A}[r'] \vdash \widetilde{B} \triangleq \mathcal{U}_k \downarrow^r_{r'} B[\widetilde{x}[r]/x] \ [i \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.B_\varepsilon[\widetilde{x}[j]/x]}]\end{array}}{\Psi \mid \Gamma \vdash \mathcal{U}_k \downarrow^r_{r'} ((x : A) \to B) \ [i \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.(x : A_\varepsilon) \to B_\varepsilon}] = (x : \widetilde{A}[r']) \to \widetilde{B} \ \mathit{type}_l}$$

EQUALITY COMPOSITION COMPUTATION

$$\frac{\Psi \mid \Gamma \vdash H \triangleq A\langle s/i\rangle \downarrow^r_{r'} P(s) \ [k \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.Q_\varepsilon(s)}]}{\Psi \mid \Gamma \vdash (\mathsf{Eq}_{i.A}(N_0, N_1) \downarrow^r_{r'} P \ [k \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.Q_\varepsilon}])(s) = H : A\langle s/i\rangle}$$

EQUALITY TYPE COMPOSITION

$$\frac{\begin{array}{c}\Psi, j, i \mid \Gamma \vdash \widetilde{A}[j, i] \triangleq \mathcal{U}_k \downarrow^r_j A \ [k \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.A_\varepsilon}] \\ \Psi \mid \Gamma \vdash \widetilde{M} \triangleq [j.\widetilde{A}[j, r]] \downarrow^r_{r'} M \ [k \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.M_\varepsilon}] \\ \Psi \mid \Gamma \vdash \widetilde{N} \triangleq [j.\widetilde{A}[j, r']] \downarrow^r_{r'} N \ [k \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.N_\varepsilon}]\end{array}}{\Psi \mid \Gamma \vdash \mathcal{U}_k \downarrow^r_{r'} \mathsf{Eq}_{i.A}(M, N) \ [k \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.\mathsf{Eq}_{i.A_\varepsilon}(M_\varepsilon, N_\varepsilon)}] = \mathsf{Eq}_{i.\widetilde{A}[r', i]}(\widetilde{M}, \widetilde{N}) \ \mathit{type}_l}$$

BOOLEAN TYPE COMPOSITION

$$\frac{}{\Psi \mid \Gamma \vdash \mathcal{U}_k \downarrow_{r'}^{r} \mathsf{bool}\ [i\ \mathsf{with}\ \overrightarrow{\varepsilon \hookrightarrow j.\mathsf{bool}}] = \mathsf{bool}\ \mathit{type}_l}$$

UNIVERSE TYPE COMPOSITION

$$\frac{}{\Psi \mid \Gamma \vdash \mathcal{U}_{k'} \downarrow_{r'}^{r} \mathcal{U}_k\ [i\ \mathsf{with}\ \overrightarrow{\varepsilon \hookrightarrow j.\mathcal{U}_k}] = \mathcal{U}_k\ \mathit{type}_l}$$

B Algebraic model theory

We begin by giving a general formulation of a category with families (cwf) which has the structure of XTT. We work in a constructive set theory extended by Grothendieck's Axiom of Universes: every set is contained in some Grothendieck Universe; this axiom induces a ordinal-indexed hierarchy of Grothendieck universes $\mathcal{V}_k : \mathbf{Set}$. Concretely, we will be using the chain of inclusions $\mathcal{V}_0 \in \ldots \in \mathcal{V}_n \in \ldots \in \mathcal{V}_\omega$.

Let $\square$ be the Cartesian cube category, the free strictly associative Cartesian category generated by an interval; concretely, its objects are dimension contexts Ψ *cube*$_+$, with morphisms given by substitutions between them. For the sake of clarity, we choose to work with the explicit syntactic presentation where Ψ is a list of named variables. Next, let $\square_+$ be the *augmented Cartesian cube category*, which freely adjoins an initial object $\perp$ to $\square$. Using the initial object, we can see that $\square_+$ has all equalizers, and that the evident functor $\square \longrightarrow \square_+$ is left exact (in other words, the new limits coincide with the old ones where they existed).

B.1 Algebraic Cumulative Cwfs

We employ a variation on the notion of categories with families (cwfs) [21] suitable for modeling dependent type theory with a hierarchy of universes à la Russell which is cumulative in an algebraic sense (i.e. without subtyping).[10]

B.1.1 Basic cwf structure: contexts, types, elements

Here, we develop the basic *judgmental* structure of a model of XTT, prior to requiring the existence of various connectives.

B.1.1.1 Category of contexts

An algebraic cumulative cwf begins with (the data of) a category $\mathcal{C}$ of *contexts* Γ, with morphisms $\Gamma \longrightarrow \Delta$ interpreting substitutions. We require there to be a terminal context $\cdot$ such that for any Γ there is a unique substitution $\Gamma \longrightarrow \cdot$.

▶ **Notation B.1** (Yoneda isomorphism)**.** For a presheaf $X : \widehat{\mathcal{C}}$ on any category $\mathcal{C}$, we will use the following notations for the components of the Yoneda isomorphism:

$$X(\Gamma) \underset{\lceil - \rceil}{\overset{\lfloor - \rfloor}{\rightleftarrows}} \widehat{\mathcal{C}}(\mathbf{y}\Gamma, X)$$

[10] We emphasize that although the data of a cwf contains the data of a category, we are doing an *algebraic* model theory for type theory in which (e.g.) the initial object is determined up to isomorphism rather than up to equivalence.

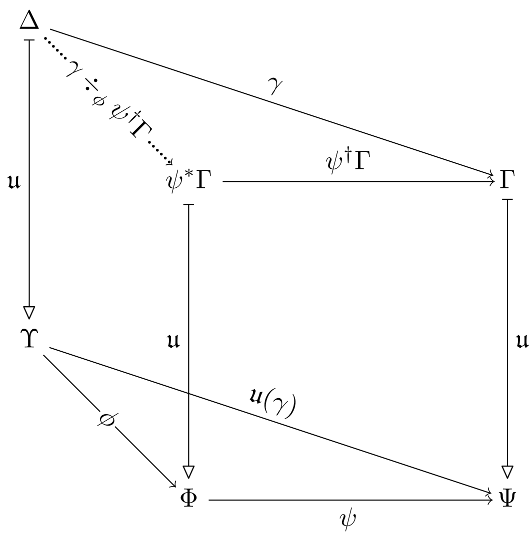


Figure 3 A schematic illustration of the situation induced by a split fibration $\mathfrak{u}$; $\psi^\dagger\Gamma$ is the Cartesian lifting of ψ, and the dotted arrow is the one induced by its universal property.

B.1.1.2 Cubical structure

We furthermore require our category of contexts be equipped with a *split* fibration $\mathcal{C} \xrightarrow{\mathfrak{u}} \square_+$. To be precise, we equip $\mathcal{C}$ with the data of a functor $\mathcal{C} \xrightarrow{|\mathfrak{u}|} \square_+$ and a splitting $(-)^\dagger$ for $|\mathfrak{u}|$. It is important to note that we intend this structure to be preserved on the nose in homomorphisms of structured cwfs.

▶ **Notation B.2** (Split fibration)**.** We impose the following notations for working with the split fibration $\mathfrak{u}$, supplemented by Figure 3.

1. The fiber of $\mathfrak{u}$ over Ψ is written $\mathcal{C}_\Psi$; explicitly, this is the subcategory of $\mathcal{C}$ whose objects are taken to Ψ by $\mathfrak{u}$.
2. The splitting $(-)^\dagger$ takes a dimension substitution $\Psi' \xrightarrow{\psi} \Psi$ and an object $\Gamma : \mathcal{C}_\Psi$ to a morphism $\psi^*\Gamma \xrightarrow{\psi^\dagger\Gamma} \Gamma$ such that $\mathfrak{u}(\psi^\dagger\Gamma) = \psi$.
3. Given a Cartesian morphism $\Delta \xrightarrow{\gamma_\Delta} \Gamma$ and an arrow $\Xi \xrightarrow{\gamma_\Xi} \Gamma$, along with an arrow in the base category $\mathfrak{u}(\Xi) \xrightarrow{\psi} \mathfrak{u}(\Gamma)$ such that $\mathfrak{u}(\gamma_\Xi) = \mathfrak{u}(\gamma_\Delta) \circ \psi$, the universal property of the Cartesian morphism guarantees a unique map $\gamma_\Xi \div_\psi \gamma_\Delta$ which lies over ψ, such that $\gamma_\Xi = \gamma_\Delta \circ (\gamma_\Xi \div_\psi \gamma_\Delta)$.
4. Given $\Delta \xrightarrow{\gamma} \Gamma$ (not necessarily vertical) and Ψ disjoint from $\mathfrak{u}(\Delta), \mathfrak{u}(\Gamma)$, we write $\hat{\Psi}^*\Delta \xrightarrow{\hat{\Psi}^+\gamma} \hat{\Psi}^*\Gamma$ for the morphism $\left(\gamma \circ \hat{\Psi}^\dagger\Delta\right) \div_{\mathfrak{u}(\gamma)\star\mathbf{id}_\Psi} \hat{\Psi}^\dagger\Gamma$ where $\mathfrak{u}(\gamma) \star \mathbf{id}_\Psi$ is the horizontal composite $\mathfrak{u}(\Delta), \Psi \to \mathfrak{u}(\Gamma), \Psi$.
5. For a presheaf $\mathcal{F} : \widehat{\mathcal{C}}$, we will write $\mathcal{F}(\Gamma) \xrightarrow{\psi^\ddagger_\Gamma} \mathcal{F}(\psi^*\Gamma)$ for the action of $\mathcal{F}$ on $\psi^*\Gamma \xrightarrow{\psi^\dagger\Gamma} \Gamma$.

B.1.1.3 Contexts and levels

$\mathcal{C}$ is the category of contexts; but the judgments of XTT are also parameterized in a universe level. Therefore, we will need to work in presheaves not on $\mathcal{C}$ but rather on $\mathcal{C}_\mathbb{L} \triangleq \mathcal{C} \times \mathbb{L}$. Recall that $\mathbb{L}$ is the category of *universe levels* and is defined by $\mathbb{L} = \omega^{\mathsf{op}}$.

B.1.1.4 Types and elements

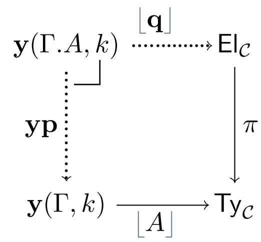
Next, we require a presheaf $\mathsf{Ty}_{\mathcal{C}}$ in $\widehat{\mathcal{C}_{\mathbb{L}}}$, yielding at each (Γ, k) the set $\mathsf{Ty}^k_{\mathcal{C}}(\Gamma)$ of types of level k over Γ; we require each fiber $\mathsf{Ty}^k_{\mathcal{C}}(\Gamma)$ to be $(k+1)$-small in the ambient set theory. As a matter of notation, we write $\Uparrow^l_k A$ for level restrictions, mirroring the concrete syntax of XTT.

Then, we require a *dependent presheaf* $\mathsf{El}_{\mathcal{C}} : \widehat{\int \mathsf{Ty}_{\mathcal{C}}}$, writing $\mathsf{El}_{\mathcal{C}}(\Gamma \vdash A)$ for the fiber over $A \in \mathsf{Ty}^k_{\mathcal{C}}(\Gamma)$; in order to justify the rules which make elements of types and their level liftings definitionally interchangeable, we require that functorial action of maps $(\Gamma, l, \Uparrow^l_k A) \longrightarrow (\Gamma, k, A)$ in $\int \mathsf{Ty}_{\mathcal{C}}$ on the dependent presheaf $\mathsf{El}_{\mathcal{C}}$ must be strict identities. Consequently, we have $\mathsf{El}_{\mathcal{C}}(\Gamma \vdash A) = \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \Uparrow^l_k A)$, following [37].

By taking a dependent sum, it is possible to regard $\mathsf{El}_{\mathcal{C}}$ as an element of the slice $\widehat{\mathcal{C}_{\mathbb{L}}}/\mathsf{Ty}_{\mathcal{C}}$; this perspective will be profitable when defining the notion of a *context comprehension*.

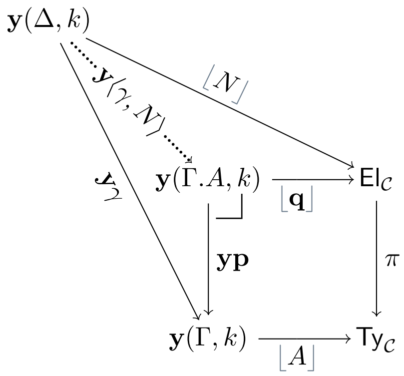
B.1.1.5 Context comprehension

Writing $\mathsf{El}_{\mathcal{C}} \xrightarrow{\pi} \mathsf{Ty}_{\mathcal{C}} : \widehat{\mathcal{C}_{\mathbb{L}}}$ for the evident projection of types from elements, we follow [23, 11] in requiring that π be equipped with a choice of representable pullbacks along natural transformations out of representable objects (contexts):



We require the equation $\Gamma.A = \Gamma.\Uparrow^l_k A$ in the objects of $\mathcal{C}$.

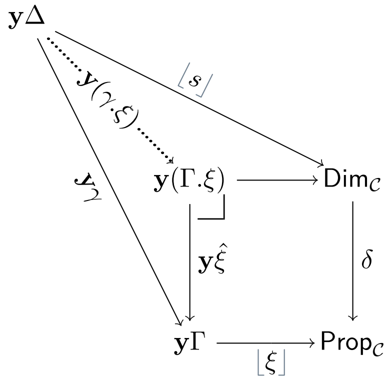
Given a substitution $\Delta \xrightarrow{\gamma} \Gamma$ and an element $N \in \mathsf{El}_{\mathcal{C}}(\Delta \vdash \gamma^* A)$, we can form the extended substitution $\langle \gamma, N \rangle$ using the universal property of the pullback:



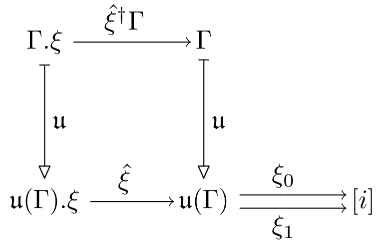
B.1.1.6 Constraint comprehension

Let $\mathsf{Dim}_{\mathcal{C}} : \widehat{\mathcal{C}}$ be the presheaf of dimensions, taking a context $\Gamma : \mathcal{C}$ to the set of dimensions $\mathsf{u}(\Gamma) \xrightarrow{r} [i]$. Then, define $\mathsf{Prop}_{\mathcal{C}} = \mathsf{Dim}_{\mathcal{C}} \times \mathsf{Dim}_{\mathcal{C}}$, writing $(r = s) \in \mathsf{Prop}_{\mathcal{C}}(\Gamma)$ for the pair of $r, s \in \mathsf{Dim}_{\mathcal{C}}(\Gamma)$; we will follow the syntax of XTT in using ξ to range over an element of $\mathsf{Prop}_{\mathcal{C}}(\Gamma)$.

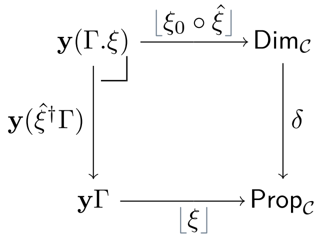
▶ **Lemma B.3.** *The split fibration* $\mathfrak{u}$ *forces the diagonal* $\mathsf{Dim}_{\mathcal{C}} \xrightarrow{\delta} \mathsf{Prop}_{\mathcal{C}}$ *to be representable in the same sense as above; schematically:*



Proof. The context $\Gamma.\xi$ is obtained from the equalizer of ξ in $\square_+$, using the splitting of the fibration $\mathfrak{u}$:



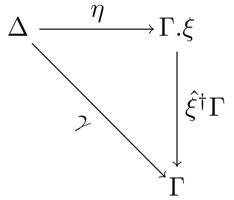
We can see that $\mathbf{y}(\Gamma.\xi)$ is indeed the pullback below:

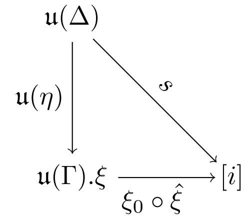


To see that the diagram commutes, we just verify that $\lfloor \xi_0 \circ \hat{\xi} = \xi_0 \circ \hat{\xi} \rfloor = \lfloor \xi \rfloor \circ \mathbf{y}(\hat{\xi}^\dagger\Gamma)$, which is the same as to say that $\xi_0 \circ \hat{\xi} = \xi_1 \circ \hat{\xi}$; but this is just the fact that $\hat{\xi}$ is the equalizer of ξ_0, ξ_1. Next, we check the universal property of the pullback; because limits in $\widehat{\mathcal{C}}$ are formed pointwise (as in all presheaf categories), it suffices to check universality at representable objects only.

Fix $\mathbf{y}\Delta \xrightarrow{\mathbf{y}\gamma} \mathbf{y}\Gamma$ and $\mathbf{y}\Delta \xrightarrow{\lfloor s \rfloor} \mathsf{Dim}_{\mathcal{C}}$ such that $\delta \circ \lfloor s \rfloor = \lfloor \xi \rfloor \circ \mathbf{y}\gamma$; we need to choose a unique morphism $\mathbf{y}\Delta \xrightarrow{\mathbf{y}\eta} \mathbf{y}\Gamma.\xi$ such that $\mathbf{y}(\xi^\dagger\Gamma) \circ \mathbf{y}\eta = \mathbf{y}\gamma$ and $\lfloor \xi_0 \rfloor \circ \mathbf{y}\eta = \lfloor s \rfloor$. Unraveling the Yoneda paperwork, we have assumed that $\xi_0 \circ \mathfrak{u}(\gamma) = s = \xi_1 \circ \mathfrak{u}(\gamma)$ and we want to find $\Delta \xrightarrow{\eta} \Gamma.\xi$ such that the following triangles commute in $\mathcal{C}$ and $\square_+$ respectively:

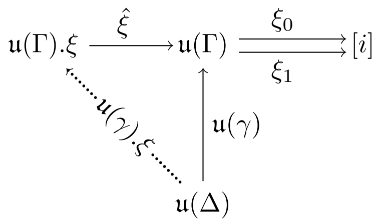
(1)



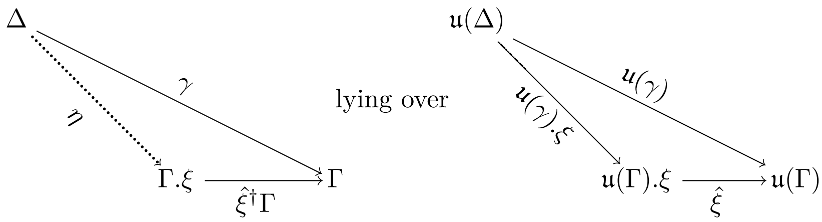


(2)

First, observe that because $\xi_0 \circ \mathfrak{u}(\gamma) = \xi_1 \circ \mathfrak{u}(\gamma)$, the universal property of the equalizer guarantees a unique map $\mathfrak{u}(\Delta) \xrightarrow{\mathfrak{u}(\gamma).\xi} \mathfrak{u}(\Gamma).\xi$ with the same property:



Using $\mathfrak{u}(\gamma).\xi$ from above, we obtain η from the universal property of the Cartesian lifting $\Gamma.\xi \xrightarrow{\hat{\xi}^\dagger\Gamma} \Gamma$:



We therefore see immediately that triangle (1) commutes; to see that triangle (2) commutes, we calculate: $\xi_0 \circ \hat{\xi} \circ \mathfrak{u}(\eta) = \xi_0 \circ \hat{\xi} \circ \mathfrak{u}(\gamma).\xi = \xi_0 \circ \mathfrak{u}(\gamma) = s$. ◀

In order to model the collapse of the typing and equality judgments under the contraint $0 = 1$ in XTT's syntax, we will *require* that the contexts $\Gamma.0 = 1$ and $\Gamma.1 = 0$ are *initial* in $\mathcal{C}$; this implies initiality in the fibration $\mathfrak{u}$, because the equalizer $\mathfrak{u}(\Gamma).0 = 1$ is the initial object in $\square_+$.

▶ **Notation B.4** (Constraint weakening)**.** Because we will use it frequently, we will often write $\Gamma.\xi \xrightarrow{\hat{\xi}} \Gamma$ for the Cartesian lifting $\hat{\xi}^\dagger\Gamma$.

▶ **Notation B.5** (Constraint lifting)**.** When $\Delta \xrightarrow{\gamma} \Gamma$, we write $\Delta.\gamma^*\xi \xrightarrow{\hat{\xi}^+\gamma} \Gamma.\xi$ for $(\gamma \circ \widehat{\gamma^*\xi}).\xi$.

We implicitly lift everything to do with dimensions and constraints into $\widehat{\mathcal{C}_{\mathbb{L}}}$, by reindexing silently along the projection $\mathcal{C}_{\mathbb{L}} \longrightarrow \mathcal{C}$.

B.1.1.7 Boundary separation

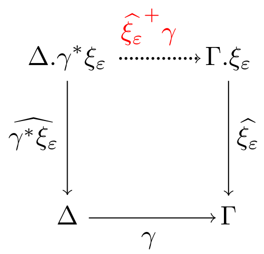
To characterize models of XTT, we need to ensure that every type and every element is totally determined by its boundary with respect to the dimension context. A simple way to state this requirement is as a *separation* condition with respect to a particular coverage on the category of contexts $\mathcal{C}$. We define the coverage $\mathbf{K}_\partial$ on $\mathcal{C}$ by taking the constraint weakenings $\{\widehat{r = \varepsilon}\}_\varepsilon$ to constitute a covering family for each dimension r:

$$\mathbf{K}_\partial(\Gamma) \ni \left\{\Gamma.r = \varepsilon \xrightarrow{\widehat{r=\varepsilon}} \Gamma\right\}_{\varepsilon \in \mathbf{2}} \qquad (r \in \mathsf{Dim}_{\mathcal{C}}(\Gamma))$$

▶ **Lemma B.6.** *The family of sets $\mathbf{K}_\partial$ is a coverage on $\mathcal{C}$.*

Proof. To see that $\mathbf{K}_\partial$ is in fact a coverage, we fix $\Delta \xrightarrow{\gamma} \Gamma$ and observe that any covering family $\{\Gamma.\xi_\varepsilon \xrightarrow{\widehat{\xi_\varepsilon}} \Gamma\}_{\varepsilon \in \mathbf{2}}$ can be pulled back to obtain a new covering family

$\{\Delta.\gamma^*\xi_\varepsilon \xrightarrow{\widehat{\gamma^*\xi_\varepsilon}} \Delta\}_{\varepsilon\in\mathbf{2}}$ such that each composite $\gamma \circ \widehat{\gamma^*\xi_\varepsilon}$ factors through $\Gamma.\xi_\varepsilon \xrightarrow{\widehat{\xi_\varepsilon}} \Gamma$:



◀

This coverage lifts immediately along the projection $\mathcal{C}_{\mathbb{L}} \longrightarrow \mathcal{C}$ to a coverage on $\mathcal{C}_{\mathbb{L}}$; because it will not result in ambiguity, we leave this lifting implicit.

▶ **Definition B.7** (Separation). *Given a coverage* $\mathbf{K}$ *on a category* $\mathcal{C}$*, a presheaf* $F : \widehat{\mathcal{C}}$ *is* $\mathbf{K}$-separated *when, for any elements* $a, b \in F(\Gamma)$ *and covering family* $\{\Delta_i \xrightarrow{\gamma_i} \Gamma\}_{i\in I} \in \mathbf{K}(\Gamma)$*, if we have* $\gamma_i^* a = \gamma_i^* b \in F(\Delta_i)$ *for each* $i \in I$*, then* $a = b \in F(\Gamma)$*.*

▶ **Definition B.8** (Boundary separation). *We say that a cwf has* boundary separation *when the presheaves* $\mathsf{Ty}_{\mathcal{C}}, \mathsf{El}_{\mathcal{C}} : \widehat{\mathcal{C}_{\mathbb{L}}}$ *are* $\mathbf{K}_\partial$*-separated.*

B.1.2 Kan operations: coercion and composition

▶ **Definition B.9** (Regular coercion structure). *A cwf has* regular coercion structure *iff for every type* $A \in \mathsf{Ty}^n_{\mathcal{C}}(\hat{\imath}^*\Gamma)$ *over* Ψ, i *and dimensions* $r, r' \in \mathsf{Dim}_{\mathcal{C}}(\Gamma)$ *and element* $M \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash (r/i)^{\ddagger}_{\hat{\imath}^*\Gamma} A)$*, there is an element* $\mathbf{coe}^{r\rightsquigarrow r'}_{i.A}\ M \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash (r'/i)^{\ddagger}_{\hat{\imath}^*\Gamma} A)$ *which has the following properties:*

- Adjacency. *If* $r = r'$ *then* $\mathbf{coe}^{r\rightsquigarrow r'}_{i.A}\ M = M$*.*
- Regularity. *If* $A = \hat{\imath}^{\ddagger}_{\Gamma} A'$ *for some* $A' \in \mathsf{Ty}^n_{\mathcal{C}}(\Gamma)$*, then* $\mathbf{coe}^{r\rightsquigarrow r'}_{i.A}\ M = M$*.*
- Level restriction. *The equation* $\mathbf{coe}^{r\rightsquigarrow r'}_{i.\Uparrow^l_k A}\ M = \mathbf{coe}^{r\rightsquigarrow r'}_{i.A}\ M$*.*
- Naturality. *For* $\Delta \xrightarrow{\gamma} \Gamma$ *we have* $\gamma^* \mathbf{coe}^{r\rightsquigarrow r'}_{i.A}\ M = \mathbf{coe}^{\gamma^* r\rightsquigarrow \gamma^* r'}_{i.(\hat{\imath}^+\gamma)^* A}\ \gamma^* M$*.*

▶ **Definition B.10** (Regular homogeneous composition structure). *We say that* $\mathcal{C}$ *has* regular homogeneous composition structure *iff, for each* $A \in \mathsf{Ty}^n_{\mathcal{C}}(\Gamma)$ *and* $r, r', s \in \mathsf{Dim}_{\mathcal{C}}(\Gamma)$ *and* $\overrightarrow{M \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash A)}$ *and* $N_\varepsilon \in \mathsf{El}_{\mathcal{C}}(\hat{\jmath}^*(\Gamma.s = \varepsilon) \vdash \hat{\jmath}^{\ddagger}\widehat{s = \varepsilon}^* A)$ *for fresh* j *such that* $\overrightarrow{(r/j)^{\ddagger} N_\varepsilon = M}$*, we have an element* $\mathbf{hcom}^{r\rightsquigarrow r'}_A\ M\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.N_\varepsilon}]$ *satisfying the following conditions:*

- Adjacency. *If* $r = r'$ *then* $\mathbf{hcom}^{r\rightsquigarrow r'}_A\ M\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.N_\varepsilon}] = M$*; moreover, if* $s = \varepsilon$*, then* $\mathbf{hcom}^{r\rightsquigarrow r'}_A\ M\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.N_\varepsilon}] = (r'/j)^{\ddagger} N_\varepsilon$*.*
- Regularity. *If we have* $\overrightarrow{N_\varepsilon = \hat{\jmath}^{\ddagger} N'_\varepsilon}$ *for some* N'*, then we have the equation* $\mathbf{hcom}^{r\rightsquigarrow r'}_A\ M\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.N_\varepsilon}] = M$*.*
- Naturality. *For* $\Delta \xrightarrow{\gamma} \Gamma$*, we require the following naturality conditions:*

$$\gamma^* \mathbf{hcom}^{r\rightsquigarrow r'}_A\ M\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.N_\varepsilon}] = \mathbf{hcom}^{\gamma^* r\rightsquigarrow \gamma^* r'}_{\gamma^* A}\ \gamma^* M\ [j.(\hat{\jmath}^+ \widehat{s = \varepsilon}^+ \gamma)^* N_\varepsilon]$$

▶ **Notation B.11** (Heterogeneous composition). When a cwf has coercion and homogeneous composition, we write its *heterogeneous composition* using the following definitional extension:

$$\mathbf{com}^{r\rightsquigarrow r'}_{i.A}\ M\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.N_\varepsilon}]$$

$$\triangleq \mathbf{hcom}^{r\rightsquigarrow r'}_{(r'/j)^{\ddagger} A}\left(\mathbf{coe}^{r\rightsquigarrow r'}_{i.A}\ M\right)\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.\mathbf{coe}^{i\rightsquigarrow r'}_{i.A}\ N_\varepsilon}]$$

B.1.3 Closure under type-theoretic connectives

▶ **Definition B.12** (Booleans). *A cwf has the* booleans *when it is equipped with the following structure:*

- Formation. *Types* $\mathbf{bool} \in \mathsf{Ty}^n_{\mathcal{C}}(\Gamma)$ *for all* Γ, n.
- Introduction. *Elements* $\mathbf{true} \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \mathbf{bool})$ *and* $\mathbf{false} \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \mathbf{bool})$.
- Elimination. *If* $C \in \mathsf{Ty}^n_{\mathcal{C}}(\Gamma.\mathbf{bool})$ *and* $M \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \mathbf{bool})$ *and* $N_0 \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \langle \mathbf{id}, \mathbf{true}\rangle^* C)$ *and* $N_1 \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \langle \mathbf{id}, \mathbf{false}\rangle^* C)$, *an element* $\mathbf{if}_C(M, N_0, N_1) \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \langle \mathbf{id}, M\rangle^* C)$.
- Computation. *The following equations:*

$$\mathbf{if}_C(\mathbf{true}, N_0, N_1) = N_0 \qquad \mathbf{if}_C(\mathbf{false}, N_0, N_1) = N_1$$

- Level restriction. *The following two equations:*

$$\Uparrow^l_k \mathbf{bool} = \mathbf{bool} \qquad \mathbf{if}_{\Uparrow^l_k C}(M, N_0, N_1) = \mathbf{if}_C(M, N_0, N_1)$$

- Naturality. *For* $\Delta \xrightarrow{\gamma} \Gamma$, *the following naturality equations:*

$$\gamma^* \mathbf{bool} = \mathbf{bool} \qquad \gamma^* \mathbf{true} = \mathbf{true} \qquad \gamma^* \mathbf{false} = \mathbf{false}$$

$$\gamma^* \mathbf{if}_C(M, N_0, N_1) = \mathbf{if}_{\langle \gamma \circ \mathbf{p}, \mathbf{q}\rangle^* C}(\gamma^* M, \gamma^* N_0, \gamma^* N_1)$$

▶ **Definition B.13** (Dependent function types). *A cwf has* dependent function types *when it is equipped with the following structure:*

- Formation. *For each type* $A \in \mathsf{Ty}^n_{\mathcal{C}}(\Gamma)$ *and family* $B \in \mathsf{Ty}^n_{\mathcal{C}}(\Gamma.A)$, *a type* $\mathbf{\Pi}(A, B) \in \mathsf{Ty}^n_{\mathcal{C}}(\Gamma)$.
- Introduction. *For each* $M \in \mathsf{El}_{\mathcal{C}}(\Gamma.A \vdash B)$, *an element* $\mathbf{lam}(M) \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \mathbf{\Pi}(A, B))$.
- Elimination. *For each* $M \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \mathbf{\Pi}(A, B))$ *and* $N \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash A)$ *an element* $\mathbf{app}(A, B, M, N) \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \langle \mathbf{id}, N\rangle^* B)$.
- Computation. *For each* $M \in \mathsf{El}_{\mathcal{C}}(\Gamma.A \vdash B)$ *and* $N \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash A)$, *the equation* $\mathbf{app}(A, B, \mathbf{lam}(M), N) = \langle \mathbf{id}, N\rangle^* M$.
- Unicity. *For each* $M \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \mathbf{\Pi}(A, B))$, *the equation* $M = \mathbf{lam}(\mathbf{app}(\mathbf{p}^* A, \langle \mathbf{p} \circ \mathbf{p}, \mathbf{q}\rangle^* B, \mathbf{p}^* M, \mathbf{q}))$.
- Level restriction. *The equation* $\Uparrow^l_k \mathbf{\Pi}(A, B) = \mathbf{\Pi}(\Uparrow^l_k A, \Uparrow^l_k B)$.
- Naturality. *We have the following naturality conditions for each* $\Delta \xrightarrow{\gamma} \Gamma$:

$$\gamma^* \mathbf{\Pi}(A, B) = \mathbf{\Pi}(\gamma^* A, \langle \gamma \circ \mathbf{p}, \mathbf{q}\rangle^* B) \qquad \gamma^* \mathbf{lam}(M) = \mathbf{lam}(\langle \gamma \circ \mathbf{p}, \mathbf{q}\rangle^* M)$$

$$\gamma^* \mathbf{app}(A, B, M, N) = \mathbf{app}(\gamma^* A, \langle \gamma \circ \mathbf{p}, \mathbf{q}\rangle^* B, \gamma^* M, \gamma^* N)$$

- Coercion. *When* $\Gamma \overset{\mathfrak{u}}{\longmapsto\!\!\rhd} \Psi, i$ *and* $\Psi \mid r, r'$ *dim and* $M \in \mathsf{El}_{\mathcal{C}}((r/i)^* \Gamma \vdash (r/i)^\ddagger \mathbf{\Pi}(A, B))$, *we require the following equation:*

$$\mathbf{coe}^{r \rightsquigarrow r'}_{i.\mathbf{\Pi}(A,B)}\, M =$$

$$\mathbf{lam}(\mathbf{coe}^{r \rightsquigarrow r'}_{i.\langle \mathbf{id}, \mathbf{coe}^{r' \rightsquigarrow i}_{i.\mathbf{p}^* A}\, \mathbf{q}\rangle^* B}\, \mathbf{app}(\mathbf{p}^* (r/i)^\ddagger A, \langle \mathbf{p} \circ \mathbf{p}, \mathbf{q}\rangle^* (r/i)^\ddagger B, \mathbf{p}^* M, \mathbf{coe}^{r' \rightsquigarrow r}_{i.\mathbf{p}^* A}\, \mathbf{q}))$$

▶ **Definition B.14** (Dependent pair types). *A cwf has* dependent pair types *when it is equipped with the following structure:*

- Formation. *For each type* $A \in \mathsf{Ty}^n_{\mathcal{C}}(\Gamma)$ *and family* $B \in \mathsf{Ty}^n_{\mathcal{C}}(\Gamma.A)$, *a type* $\mathbf{\Sigma}(A, B) \in \mathsf{Ty}^n_{\mathcal{C}}(\Gamma)$.
- Introduction. *For each* $M \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash A)$ *and* $N \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \langle \mathbf{id}, M\rangle^* B)$, *an element* $\mathbf{pair}(M, N) \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \mathbf{\Sigma}(A, B))$.

- Elimination. *For each* $M \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \mathbf{\Sigma}(A,B))$*, elements* $\mathbf{fst}(A,B,M) \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash A)$ *and* $\mathbf{snd}(A,B,M) \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash [\mathbf{id}, \mathbf{fst}(A,B,M)]^*B)$.
- Computation. *For* $M \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash A)$ *and* $N \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash [\mathbf{id}, M]^*B)$*, the following equations:*

$$\mathbf{fst}(A,B,\mathbf{pair}(M,N)) = M \qquad \mathbf{snd}(A,B,\mathbf{pair}(M,N)) = N$$

- Unicity. *For* $M \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \mathbf{\Sigma}(A,B))$*, the equation* $M = \mathbf{pair}(\mathbf{fst}(A,B,M), \mathbf{snd}(A,B,M))$.
- Level restriction. *The equation* $\Uparrow_k^l \mathbf{\Sigma}(A,B) = \mathbf{\Sigma}(\Uparrow_k^l A, \Uparrow_k^l B)$.
- Naturality. *For substitutions* $\Delta \xrightarrow{\gamma} \Gamma$ *the following naturality equations:*

$$\gamma^*\mathbf{\Sigma}(A,B) = \mathbf{\Sigma}(\gamma^*A, \langle \gamma \circ \mathbf{p}, \mathbf{q}\rangle^*B) \qquad \gamma^*\mathbf{pair}(M,N) = \mathbf{pair}(\gamma^*M, \gamma^*N)$$

$$\gamma^*\mathbf{fst}(A,B,M) = \mathbf{fst}(\gamma^*A, \langle \gamma \circ \mathbf{p}, \mathbf{q}\rangle^*B, \gamma^*M)$$

$$\gamma^*\mathbf{snd}(A,B,M) = \mathbf{snd}(\gamma^*A, \langle \gamma \circ \mathbf{p}, \mathbf{q}\rangle^*B, \gamma^*M)$$

- Coercion. *When* $\Gamma \overset{\mathfrak{u}}{\longmapsto} \Psi, i$ *and* $\Psi \mid r, r'$ *dim and* $M \in \mathsf{El}_{\mathcal{C}}((r/i)^*\Gamma \vdash (r/i)^\ddagger \mathbf{\Sigma}(A,B))$*, we require the following equation:*

$$\frac{\begin{array}{l} M_0 \triangleq \mathbf{coe}_{i.A}^{r \rightsquigarrow r'}\ \mathbf{fst}((r/i)^\ddagger A, (r/i)^\ddagger B, M) \\ M_1 \triangleq \mathbf{coe}^{r \rightsquigarrow r'}_{i.\langle \mathbf{id}, \mathbf{coe}_{i.A}^{r \rightsquigarrow i}\ \mathbf{fst}((r/i)^\ddagger A, (r/i)^\ddagger B, M)\rangle^* B}\ \mathbf{snd}((r/i)^\ddagger A, (r/i)^\ddagger B, M) \end{array}}{\mathbf{coe}^{r \rightsquigarrow r'}_{i.\mathbf{\Sigma}(A,B)}\ M = \mathbf{pair}(M_0, M_1)}$$

▶ **Definition B.15** (Dependent path types)**.** *A cwf has* dependent path types *when it is equipped with the following structure:*

- Formation. *For each type* $A \in \mathsf{Ty}^n_{\mathcal{C}}(\hat{\imath}^*\Gamma)$ *and elements* $\overrightarrow{N_\varepsilon \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash (\varepsilon/i)^\ddagger A)}$*, a type* $\mathbf{Path}(i.A, N_0, N_1) \in \mathsf{Ty}^n_{\mathcal{C}}(\Gamma)$.
- Introduction. *For each* $M \in \mathsf{El}_{\mathcal{C}}(\hat{\imath}^*\Gamma \vdash A)$*, an element* $\mathbf{plam}(M) \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \mathbf{Path}(i.A, (0/i)^\ddagger M, (1/i)^\ddagger M))$.
- Elimination. *For each* $M \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \mathbf{Path}(i.A, N_0, N_1))$ *and* $r \in \mathsf{Dim}_{\mathcal{C}}(\Gamma)$*, an element* $\mathbf{papp}(i.A, M, r) \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash (r/i)^\ddagger A)$ *satisfying the equations* $\overrightarrow{\mathbf{papp}(i.A, M, \epsilon) = N_\epsilon}$.
- Computation. *For* $M \in \mathsf{El}_{\mathcal{C}}(\hat{\imath}^*\Gamma \vdash A)$ *and* $r \in \mathsf{Dim}_{\mathcal{C}}(\Gamma)$*, the equation* $\mathbf{papp}(i.A, \mathbf{plam}(i.M), r) = (r/i)^\ddagger M$.
- Boundary. *For* $M \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \mathbf{Path}(i.A, N_0, N_1))$*, the equations* $\mathbf{papp}(i.A, M, 0) = N_0$ *and* $\mathbf{papp}(i.A, M, 1) = N_1$.
- Unicity. *For* $M \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \mathbf{Path}(i.A, N_0, N_1))$*, the equation* $M = \mathbf{plam}(j.\mathbf{papp}(i.\hat{\jmath}^\ddagger A, \hat{\jmath}^\ddagger M, j))$.
- Level restriction. *The equation* $\Uparrow_k^l \mathbf{Path}(i.A, N_0, N_1) = \mathbf{Path}(i.\Uparrow_k^l A, N_0, N_1)$.
- Naturality. *For* $\Delta \xrightarrow{\gamma} \Gamma$*, the following naturality equations:*

$$\gamma^*\mathbf{Path}(i.A, N_0, N_1) = \mathbf{Path}(i.(\hat{\imath}^+\gamma)^*A, \gamma^*N_0, \gamma^*N_1)$$

$$\gamma^*\mathbf{plam}(i.M) = \mathbf{plam}(i.(\hat{\imath}^+\gamma)^*M) \qquad \gamma^*\mathbf{papp}(i.A, M, r) = \mathbf{papp}(i.(\hat{\imath}^+\gamma)^*A, \gamma^*M, \gamma^*r)$$

- Coercion. *When* $\Gamma \overset{\mathfrak{u}}{\longmapsto} \Psi, j$ *and* $\Psi \mid r, r'$ *dim and* $M \in \mathsf{El}_{\mathcal{C}}((r/j)^*\Gamma \vdash (r/j)^\ddagger \mathbf{Path}(i.A, N_0, N_1))$*, we require the following equation:*

$$\mathbf{coe}^{r \rightsquigarrow r'}_{j.\mathbf{Path}(i.A,N_0,N_1)}\ M = \mathbf{plam}(i.\mathbf{com}^{r \rightsquigarrow r'}_{j.A}\ \mathbf{papp}(i.(r/j)^\ddagger A, \hat{\imath}^\ddagger M, i)\ [i \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.\hat{\jmath}^\ddagger N_\varepsilon}])$$

▶ **Definition B.16** (Dependent equality types)**.** *A cwf which has both boundary separation and dependent path types is said to have* dependent equality types*, and we accordingly write* $\mathbf{Eq}(i.A, M, N)$ *for* $\mathbf{Path}(i.A, M, N)$.

▶ **Definition B.17** (Universes à la Russell). *An algebraic cumulative cwf has universes à la Russell iff for all levels $k < l$ and context $\Gamma : \mathcal{C}$, there is a type $\mathbf{U}_k \in \mathsf{Ty}^l_{\mathcal{C}}(\Gamma)$ such that $\mathsf{El}_{\mathcal{C}}(\Gamma \vdash \mathbf{U}_k) = \mathsf{Ty}^k_{\mathcal{C}}(\Gamma)$. We additionally require the naturality equations $\gamma^*\mathbf{U}_k = \mathbf{U}_k$ and $\Uparrow^m_l \mathbf{U}_k = \mathbf{U}_k$.*

▶ **Definition B.18** (Type-case). *An algebraic cumulative cwf has* type-case *iff given the following data,*

$$C \in \mathsf{Ty}^l_{\mathcal{C}}(\Gamma) \qquad X \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \mathbf{U}_k) \qquad M_\Pi, M_\Sigma \in \mathsf{El}_{\mathcal{C}}(\Gamma.\mathbf{U}_k.\mathbf{\Pi}(\mathbf{q}, \mathbf{U}_k) \vdash (\mathbf{p} \circ \mathbf{p})^* C)$$

$$M_{\mathbf{Eq}} \in \mathsf{El}_{\mathcal{C}}(\Gamma.\mathbf{U}_k.\mathbf{U}_k.\mathbf{Eq}(\_.\mathbf{U}_k, \mathbf{p}^*\mathbf{q}, \mathbf{q}).(\mathbf{p} \circ \mathbf{p})^*\mathbf{q}.(\mathbf{p} \circ \mathbf{p})^*\mathbf{q} \vdash (\mathbf{p} \circ \mathbf{p} \circ \mathbf{p} \circ \mathbf{p} \circ \mathbf{p})^* C)$$

$$M_{\mathbf{bool}} \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash C) \qquad M_{\mathbf{U}} \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash C)$$

we have an element $\mathbf{case}_{\mathbf{U}_k}(C; X; M_\Pi; M_\Sigma; M_{\mathbf{Eq}}; M_{\mathbf{bool}}; M_{\mathbf{U}}) \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash C)$ such that the following conditions hold:

- Computation.

$$\mathbf{case}_{\mathbf{U}_k}(C; \mathbf{\Pi}(A, B); M_\Pi; M_\Sigma; M_{\mathbf{Eq}}; M_{\mathbf{bool}}; M_{\mathbf{U}}) = \langle\langle \mathbf{id}, A\rangle, \mathbf{lam}(B)\rangle^* M_\Pi$$

$$\mathbf{case}_{\mathbf{U}_k}(C; \mathbf{\Sigma}(A, B); M_\Pi; M_\Sigma; M_{\mathbf{Eq}}; M_{\mathbf{bool}}; M_{\mathbf{U}}) = \langle\langle \mathbf{id}, A\rangle, \mathbf{lam}(B)\rangle^* M_\Sigma$$

$$\frac{\eta \triangleq \langle\langle\langle\langle \mathbf{id}, (0/i)^\ddagger A\rangle, (1/i)^\ddagger A\rangle, \mathbf{plam}(i.A)\rangle, N_0\rangle, N_1\rangle}{\mathbf{case}_{\mathbf{U}_k}(C; \mathbf{Eq}(i.A, N_0, N_1); M_\Pi; M_\Sigma; M_{\mathbf{Eq}}; M_{\mathbf{bool}}; M_{\mathbf{U}}) = \eta^* M_{\mathbf{Eq}}}$$

$$\mathbf{case}_{\mathbf{U}_k}(C; \mathbf{bool}; M_\Pi; M_\Sigma; M_{\mathbf{Eq}}; M_{\mathbf{bool}}; M_{\mathbf{U}}) = M_{\mathbf{bool}}$$

$$\mathbf{case}_{\mathbf{U}_k}(C; \mathbf{U}_l; M_\Pi; M_\Sigma; M_{\mathbf{Eq}}; M_{\mathbf{bool}}; M_{\mathbf{U}}) = M_{\mathbf{U}}$$

- Naturality. *For $\Delta \xrightarrow{\gamma} \Gamma$, the following naturality condition:*

$$\frac{\gamma_{+2} \triangleq \langle\langle \gamma \circ \mathbf{p} \circ \mathbf{p}, \mathbf{q}\rangle, \mathbf{q}\rangle \qquad \gamma_{+5} \triangleq \langle\langle\langle\langle\langle \gamma \circ \mathbf{p} \circ \mathbf{p} \circ \mathbf{p} \circ \mathbf{p} \circ \mathbf{p}, \mathbf{q}\rangle, \mathbf{q}\rangle, \mathbf{q}\rangle, \mathbf{q}\rangle, \mathbf{q}\rangle}{\begin{array}{l}\gamma^*\mathbf{case}_{\mathbf{U}_k}(C; X; M_\Pi; M_\Sigma; M_{\mathbf{Eq}}; M_{\mathbf{bool}}; M_{\mathbf{U}}) = \\ \mathbf{case}_{\mathbf{U}_k}(\gamma^* C; \gamma^* X; \gamma^*_{+2} M_\Pi; \gamma^*_{+2} M_\Sigma; \gamma^*_{+5} M_{\mathbf{Eq}}; \gamma^* M_{\mathbf{bool}}; \gamma^* M_{\mathbf{U}})\end{array}}$$

B.2 Syntactic model and initiality

The cwfs with all the structure described in Appendix B can be arranged into a category which has an initial object. This is because every piece of structure that we have defined in Appendix B is *generalized algebraic* in the sense of [14]; even the universe structure can be seen to be generalized algebraic [37]. We conjecture (but do not prove) that the syntax of XTT can be used to construct a cwf (the Lindenbaum-Tarski algebra) which has the universal property of the initial object.

Syntactic presentation of augmented cubes

The syntactic contexts Ψ can be viewed as a particular syntactic presentation of the category $\square_+$ of augmented Cartesian cubes, in which the equalizers are implemented formally by extending Ψ with equations.

Category of contexts

The well-typed term contexts $\Psi \mid \Gamma\ ctx$ can be organized (up to formal equality) into a category with morphisms $(\Psi' \mid \Gamma') \xrightarrow{(\psi,\gamma)} (\Psi \mid \Gamma)$ with $[\![\Psi']\!] \xrightarrow{\psi} [\![\Psi]\!]$ and $\Gamma' \xrightarrow{\gamma} \psi^*\Gamma$, i.e. a well-typed substitution of terms in context $\psi^*\Gamma$ for the variables from Γ'.

Types and terms

The presheaves of types $\mathsf{Ty}^n_{\mathcal{C}}$ are given by syntactic types $\Psi \mid \Gamma \vdash A\ type_n$ up to formal equality, with action given by substitutions; well-typed terms $\Psi \mid \Gamma \vdash M : A$ taken up to formal equality generate the fibers of a natural transformation $\mathsf{El}_{\mathcal{C}} \xrightarrow{\pi} \mathsf{Ty}_{\mathcal{C}}$. The representability of π is immediate from the fact that syntactic contexts $\Psi \mid \Gamma$ can be extended by any type to yield $\Psi \mid \Gamma, x : A$. This context comes equipped with a projection $\Psi \mid \Gamma$ and $\Psi \mid \Gamma, x : A \vdash x : A$ which implements the desired pullback square. The initiality of $\Psi, 0 = 1 \mid \Gamma$ is ensured by the FALSE CONSTRAINT rule (see Appendix A.2.4).

Cubical judgmental structure

The functor $\mathcal{C} \xrightarrow{|\mathfrak{u}|} \square_+$ takes a context $\Psi \mid \Gamma$ to Ψ, and projects out the dimension component of a substitution (ψ, γ). It is equipped with a splitting which, for the context $\Psi \mid \Gamma$, lifts $\Psi' \xrightarrow{\psi} \Psi$ to $\Psi' \mid \psi^*\Gamma \xrightarrow{(\psi,\mathbf{id})} \Psi \mid \Gamma$. Semantic boundary separation is obtained immediately from the boundary separation rules.

Connectives

We observe that our cwf also has dependent function, pair, equality, boolean and universe types given by the syntax.

C Cubical logical families and gluing

We fix an *arbitrary* structured cwf $\mathcal{C}$; we will write $\Gamma : \mathcal{C}$ for its objects. We will show how to build a cwf $\mathcal{C}^\bullet$ of cubical logical families over $\mathcal{C}$, called the "computability cwf", which has some (but not all) of the structure of a model of XTT. Then, in Appendix D we will construct a genuine model $\mathcal{C}^\star$ of XTT by restricting $\mathcal{C}^\bullet$ to a closed universe.

C.1 The cubical nerve

We have a functor $\square_+ \xrightarrow{\langle - \rangle} \mathcal{C}$ which takes a dimension context to $\mathcal{C}$-context with those dimensions but no term variables:

$$\langle \Psi \rangle = \hat{\Psi}^*(\cdot)$$
$$\langle \Phi \xrightarrow{\psi} \Psi \rangle = \langle \Phi \rangle \xrightarrow{\hat{\Phi}^\dagger(\cdot) \div_\psi \hat{\Psi}^\dagger(\cdot)} \langle \Psi \rangle$$

This functor $\langle - \rangle$ induces a *nerve* construction $\mathcal{C} \xrightarrow{(\!|-|\!)} \widehat{\square}_+$, taking Γ to the presheaf $\mathcal{C}(\langle - \rangle, \Gamma)$. Intuitively, this is the presheaf of substitutions which are closed with respect to term variables, but open with respect to dimension variables; from the perspective of inside $\widehat{\square}_+$, these are the closed substitutions.

Abusing notation slightly, we now the cubical set of "closed types" $(\!|\mathsf{Ty}_k|\!)$ as $\mathsf{Ty}^k_{\mathcal{C}} \circ \langle - \rangle^{\mathsf{op}}$. We furthermore define a dependent cubical set $(\!|\mathsf{El}_k|\!)$ over $(\!|\mathsf{Ty}_k|\!)$, taking $(\Psi, A) \in \int (\!|\mathsf{Ty}_k|\!)$ to

$\mathsf{El}_{\mathcal{C}}(\langle\Psi\rangle \vdash A)$; internally, we will (abusively) write $(\!|A|\!)$ for the fiber of $(\!|\mathsf{El}_k|\!)$ over $A : (\!|\mathsf{Ty}_k|\!)$. Given $\gamma : (\!|\Gamma|\!)$ and $A \in \mathsf{Ty}^k_{\mathcal{C}}(\Gamma)$, we abuse notation by writing $\gamma^*A : (\!|\mathsf{Ty}_k|\!)$.

Furthermore, we also define a dependent cubical set $(\!|\mathsf{Fam}_n|\!)[A]$ over $A : (\!|\mathsf{Ty}_n|\!)$, which internalizes the *families* of $\mathcal{C}$-types indexed in a given $\mathcal{C}$-type. Explicitly, we define a presheaf $(\!|\mathsf{Fam}_n|\!)$ whose fibers are $\coprod_{A\in\mathsf{Ty}^n_{\mathcal{C}}(\langle\Psi\rangle)} \mathsf{Ty}^n_{\mathcal{C}}(\langle\Psi\rangle.A)$ for each Ψ, and then exhibit the obvious projection $(\!|\mathsf{Fam}_n|\!) \twoheadrightarrow (\!|\mathsf{Ty}_n|\!)$.

▶ **Lemma C.1.** *For each level n, we have $(\!|\mathbf{U}_n|\!) = (\!|\mathsf{Ty}_n|\!)$.*

Proof. This follows from the fact that $\mathcal{C}$ is a model of universes à la Russell. Calculate:

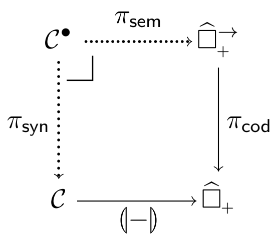
$$\begin{aligned}(\!|\mathbf{U}_n|\!) &= \mathsf{El}_{\mathcal{C}}(\langle-\rangle \vdash \mathbf{U}_n) && \text{(definition)}\\ &= \mathsf{Ty}^n_{\mathcal{C}}(\langle-\rangle) && \text{($\mathcal{C}$ has universes)}\\ &= (\!|\mathsf{Ty}_n|\!) && \text{(definition)}\end{aligned}$$

◀

C.2 Logical families by semantic gluing

By gluing the family fibration along the nerve functor $\mathcal{C} \xrightarrow{(\!|-|\!)} \widehat{\square}_+$, we acquire a category of *cubical logical families*, which we can used to prove canonicity for closed terms, instantiating $\mathcal{C}$ with the initial structured cwf. Intuitively, the role of the gluing category is to "cut down" the morphisms in cubical sets to those which are definable in $\mathcal{C}$, allowing us to extract non-trivial theorems about $\mathcal{C}$ using the very powerful tools afforded by the topos $\widehat{\square}_+$.

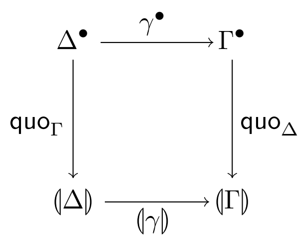
We will prefer a more explicit and type-theoretic presentation of the gluing category, but it is helpful for intuition to view it as a pullback of the fundamental fibration for $\widehat{\square}_+$ along the cubical nerve functor:



Another view of the gluing category comes from the comma construction $\mathbf{id}_{\widehat{\square}_+} \downarrow (\!|-|\!)$.

Diagrammatic construction of $\mathcal{C}^\bullet$

Explicitly, an object in $\mathcal{C}^\bullet$ is a triple $\overline{\Gamma} = (\Gamma, \Gamma^\bullet, \mathsf{quo}_\Gamma)$ of a context $\Gamma : \mathcal{C}$, a cubical set $\Gamma^\bullet$, and a natural transformation $\Gamma^\bullet \xrightarrow{\mathsf{quo}_\Gamma} (\!|\Gamma|\!)$; a morphism $\overline{\Delta} \xrightarrow{\overline{\gamma}} \overline{\Gamma}$ is then a pair $\overline{\gamma} = (\gamma, \gamma^\bullet)$ together with a commuting square:



Type-theoretic construction of $\mathcal{C}^\bullet$

Following [20], we will prefer a *type-theoretic* presentation of $\mathcal{C}^\bullet$ in terms of the hierarchy of Grothendieck universes $\mathcal{V}_n$, which lift directly into $\widehat{\square}_+$ as in [26]. We found that this type-theoretic style scales more easily to the complex situations involved in the semantics of dependent type theory than the diagrammatic style above.

According to the type-theoretic presentation, an object of $\mathcal{C}^\bullet$ is a pair $\overline{\Gamma} = (\Gamma, \Gamma^\bullet)$ with $\Gamma : \mathcal{C}$ and $\Gamma^\bullet$ a family $(\!|\Gamma|\!) \to \mathcal{V}_n$ for some n. A morphism $\overline{\Delta} \xrightarrow{\overline{\gamma}} \overline{\Gamma}$ is a pair $\overline{\gamma} = \left(\Delta \xrightarrow{\gamma} \Gamma, \gamma^\bullet\right)$ with $\gamma^\bullet : \prod_{\delta : (\!|\Delta|\!)} \Delta^\bullet \delta \to \Gamma^\bullet(\gamma^* \delta)$. To be precise, $\gamma^\bullet$ is a global element of the dependent function type in $\widehat{\square}_+$; $\gamma^\bullet$ witnesses the fact that the syntactic substitution γ preserves the logical family.

There is a slight mismatch with the earlier diagrammatic intuition: the type-theoretic presentation only allows for families whose fibers fit into $\mathcal{V}_n$ for some n. Since we will work exclusively with the more restrictive type-theoretic presentation from now on this poses no technical challenges. Those who prefer the intuition provided by the diagrammatic presentation need merely restrict the pullback construction to certain suitably small cubical sets.

What's it for?

$\Gamma^\bullet$ is a *proof-relevant* logical predicate ("logical family") on elements of Γ which may have free dimension variables, but which commutes with all substitutions of those dimension variables. In other words, $\Gamma^\bullet$ is a (cubical, proof-relevant) predicate on the elements of Γ.

C.3 Cwf structure: types and elements

A *glued type* of level l in context $\overline{\Gamma}$ is a pair $\overline{A} = (A, A^\bullet)$ with $A \in \mathsf{Ty}^l_{\mathcal{C}}(\Gamma)$ and $A^\bullet$ a global element of the cubical set $\prod_{\gamma : (\!|\Gamma|\!)} \prod_{\gamma^\bullet : \Gamma^\bullet \gamma} (\!|\gamma^* A|\!) \to \mathcal{V}_l$. Level restrictions $\Uparrow^l_k \overline{A}$ are inherited directly from $\mathcal{C}$: the family part of a glued type can remain unchanged because $(\!|A|\!) = (\!|\Uparrow^l_k A|\!)$ and $\mathcal{V}_k \subseteq \mathcal{V}_l$. A *glued element* in context $\overline{\Gamma}$ of type $\overline{A} \in \mathsf{Ty}^n_{\mathcal{C}^\bullet}(\overline{\Gamma})$ is a pair $\overline{M} = (M, M^\bullet)$ with $M \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash A)$ and $M^\bullet$ a global element of the cubical set $\prod_{\gamma : (\!|\Gamma|\!)} \prod_{\gamma^\bullet : \Gamma^\bullet \gamma} A^\bullet \gamma \gamma^\bullet (\gamma^* M)$.

We observe that the induced projection $\mathsf{El}_{\mathcal{C}^\bullet} \xrightarrow{\pi} \mathsf{Ty}_{\mathcal{C}^\bullet}$ is representable by exhibiting the evident context comprehension $\overline{\Gamma}.\overline{A}$ for $\overline{\Gamma} : \mathcal{C}^\bullet$ and $\overline{A} \in \mathsf{Ty}_{\mathcal{C}^\bullet}(\Gamma)$, taking $\Gamma.A$ for its syntactic part, and using the following family for its semantic part:

$$\left(\overline{\Gamma}.\overline{A}\right)^\bullet \langle \gamma, a \rangle = \textstyle\sum_{\gamma^\bullet : \Gamma^\bullet \gamma} A^\bullet \gamma \gamma^\bullet a$$

We clearly have that the restrictions $\mathsf{El}_{\mathcal{C}^\bullet}(\overline{\Gamma} \vdash \overline{A}) \longrightarrow \mathsf{El}_{\mathcal{C}^\bullet}(\overline{\Gamma} \vdash \Uparrow^l_k \overline{A})$ are identities, and $\overline{\Gamma}.\overline{A} = \overline{\Gamma}.\Uparrow^l_k \overline{A}$. Therefore, $\mathcal{C}^\bullet$ forms an algebraic cumulative cwf; we will call it the "computability cwf". The cubical structure is inherited from $\mathcal{C}$ by precomposing with the fibration $\mathcal{C}^\bullet \xrightarrow{\pi_{\mathsf{syn}}} \mathcal{C}$. We define the glued constraint comprehension $\overline{\Gamma}.\xi$ by taking $\Gamma.\xi$ for the syntactic part, and defining its logical family as follows:

$$\left(\overline{\Gamma}.r = s\right)^\bullet (\gamma.r = s) = \{\Gamma^\bullet \gamma \mid r = s\}$$

This choice of realizers for the constraint comprehension ensures the initiality of inconsistent contexts in the gluing model.

D Canonicity for XTT: the computability model

We have not succeeded in closing the computability cwf from Appendix C under Kan universes à la Russell of Kan types; the essential difficulties are the separation property and the regular coercion structure. Therefore, this cwf does not have the structure of a model of XTT.

To rectify this, we will restrict $\mathcal{C}^\bullet$ to a smaller cwf $\mathcal{C}^\star$, in which the types are generated inductively in a way reminiscent of the construction of closed universes in PER models [2]; the main difference is that, rather than using large induction-recursion (which has not been shown to exist in presheaf toposes), we model n object universes in the cubical universe $\mathcal{V}_{n+1}$ (an instance of *small* induction-recursion, which can be translated to constructs available in every presheaf topos [24]).

Finally, using the universal property of the type structure of the restricted cwf, we will generate the coercion and composition structure recursively, obtaining a model of XTT.

D.1 Closed universe hierarchy

We will define a family $\mathfrak{U}^\bullet_n : (\!|\mathsf{Ty}_n|\!) \to \mathcal{V}_{n+1}$ internally to $\widehat{\square}_+$, together with $(-)^\circ : \prod_{\mathtt{A}:\mathfrak{U}^\bullet_n A} (\!|A|\!) \to \mathcal{V}_n$. The former will serve as the computability predicate for a closed universe, and we will use the latter in order to define a family of types to decode the closed universe.

$$\frac{(j < n)}{\mathtt{univ}_j : \mathfrak{U}^\bullet_n \mathbf{U}_j} \qquad \frac{}{\mathtt{bool} : \mathfrak{U}^\bullet_n \mathbf{bool}}$$

$$\frac{\mathtt{A} : \mathfrak{U}^\bullet_n a \qquad \mathtt{B} : \mathfrak{F}_n[\mathtt{A}]^\bullet B}{\mathtt{pi}(\mathtt{A};\mathtt{B}) : \mathfrak{U}^\bullet_n \mathbf{\Pi}(A,B)} \qquad \frac{\mathtt{A} : \mathfrak{U}^\bullet_n A \qquad \mathtt{B} : \mathfrak{F}_n[\mathtt{A}]^\bullet B}{\mathtt{sg}(\mathtt{A};\mathtt{B}) : \mathfrak{U}^\bullet_n \mathbf{\Sigma}(A,B)} \qquad \frac{\mathtt{A} : \prod_{i:\mathbb{I}} \mathfrak{U}^\bullet_n A_i \qquad \overrightarrow{N^\bullet_\varepsilon : \mathtt{A}(\varepsilon)^\circ N_\varepsilon}}{\mathtt{eq}(\mathtt{A}; N^\bullet_0, N^\bullet_1) : \mathfrak{U}^\bullet_n \mathbf{Eq}(i.A_i, N_0, N_1)}$$

We define an auxiliary family of types to capture family of type-codes:

$$\begin{aligned}
&\mathfrak{F}_n[-]^\bullet : \textstyle\prod_{\mathtt{A}:\mathfrak{U}^\bullet_n A} (\!|\mathsf{Fam}_n|\!)[A] \to \mathcal{V}_{n+1} \\
&\mathfrak{F}_n[\mathtt{A}]^\bullet B = \textstyle\prod_{M:(\!|A|\!)} \mathtt{A}^\circ M \to \mathfrak{U}^\bullet_n(\langle \mathbf{id}, M\rangle^* B)
\end{aligned}$$

The assignment of computability families to type codes is as follows:

$$\begin{aligned}
(-)^\circ &: \textstyle\prod_{\mathtt{A}:\mathfrak{U}^\bullet_n A} (\!|A|\!) \to \mathcal{V}_n \\
\mathtt{univ}^\circ_i &= \mathfrak{U}^\bullet_i \\
\mathtt{bool}^\circ &= \lambda M.\, (M = \mathbf{true}) + (M = \mathbf{false}) \\
\mathtt{pi}(\mathtt{A};\mathtt{B})^\circ &= \lambda M.\, \textstyle\prod_{N:(\!|A|\!)} \prod_{N^\bullet:\mathtt{A}^\bullet N} (\mathtt{B} N N^\bullet)^\circ \mathbf{app}(A, B, M, N) \\
\mathtt{sg}(\mathtt{A};\mathtt{B})^\circ &= \lambda M.\, \textstyle\sum_{M^\bullet_0:\mathtt{A}^\circ \mathbf{fst}(A,B,M)} (\mathtt{B}(\mathbf{fst}(A,B,M)) M^\bullet_0)^\circ \mathbf{snd}(A,B,M) \\
\mathtt{eq}(\mathtt{A}; N^\bullet_0, N^\bullet_1)^\circ &= \lambda M.\, \left\{ M^\bullet : \textstyle\prod_{i:\mathbb{I}} \mathtt{A}(i)^\circ \mathbf{papp}(i.A, M, i) \mid \overrightarrow{M^\bullet(\varepsilon) = N^\bullet_\varepsilon} \right\}
\end{aligned}$$

▶ **Lemma D.1.** *For any $A : (\!|\mathsf{Ty}_k|\!)$ and with $k \le l$, we have $\mathfrak{U}^\bullet_k A = \mathfrak{U}^\bullet_l \Uparrow^l_k A$.*

Proof. We will show that $\mathfrak{U}^\bullet_k A \subseteq \mathfrak{U}^\bullet_l \Uparrow^l_k A$; the other direction is symmetric. Fix $\mathtt{A} : \mathfrak{U}^\bullet_k A$; we verify that $\mathtt{A} : \mathfrak{U}^\bullet_l \Uparrow^l_k A$ as well, proceeding by induction.
Case.

$$\frac{(j < k)}{\mathtt{univ}_j : \mathfrak{U}^\bullet_k \mathbf{U}_j}$$

We have $\mathtt{univ}_j : \mathfrak{U}^\bullet_l \mathbf{U}_j$, and $\mathbf{U}_j = \Uparrow^l_k \mathbf{U}_j$.

Case.

$$\frac{}{\texttt{bool} : \mathfrak{U}^\bullet_k\mathbf{bool}}$$

We likewise have $\texttt{bool} : \mathfrak{U}^\bullet_l\mathbf{bool}$, and $\mathbf{bool} = \Uparrow^l_k\mathbf{bool}$.

Case.

$$\frac{\texttt{A} : \mathfrak{U}^\bullet_k A \qquad \texttt{B} : \mathfrak{F}_k[\texttt{A}]^\bullet B}{\texttt{pi}(\texttt{A};\texttt{B}) : \mathfrak{U}^\bullet_k\mathbf{\Pi}(A,B)}$$

To see that $\texttt{pi}(\texttt{A};\texttt{B}) : \mathfrak{U}^\bullet_l\Uparrow^l_k\mathbf{\Pi}(A,B)$, by calculation, it suffices to show that $\texttt{pi}(\texttt{A};\texttt{B}) : \mathfrak{U}^\bullet_l\mathbf{\Pi}(\Uparrow^l_k A, \Uparrow^l_k B)$. By induction, we have $\texttt{A} : \mathfrak{U}^\bullet_l\Uparrow^l_k A$; to verify that $\texttt{B} : \mathfrak{F}_l[\texttt{A}]^\bullet\Uparrow^l_k B$, we fix $M : (\!|A|\!)$ and $M^\bullet : \texttt{A}^\circ M$ and need to check that $\texttt{B}MM^\bullet : \mathfrak{U}^\bullet_l\Uparrow^l_k(\langle\mathbf{id}, M\rangle^* B)$; but this follows from our second induction hypothesis and the fact that level restriction commutes with substitution.

The remaining cases are analogous. ◀

▶ **Notation D.2.** We will write $\partial(r)$ for the formula $(r = 0) \vee (r = 1)$, the *boundary* of r.

▶ **Lemma D.3.** *Internally to $\widehat{\square}_+$, the following formulas are true:*

$$\forall r : \mathbb{I}, n : \mathbb{N}, A, B : (\!|\mathsf{Ty}_n|\!).\, (\partial(r) \implies A = B) \implies A = B \qquad \text{(types)}$$

$$\forall r : \mathbb{I}, A : (\!|\mathsf{Ty}_n|\!), M, N : (\!|A|\!).\, (\partial(r) \implies M = N) \implies A = B \qquad \text{(elements)}$$

Proof. This follows from the fact that, as a model of XTT, $\mathcal{C}$ has boundary separation; therefore, its types and elements are separated with respect to ∂. This implies that all elements are completely defined by their boundaries. We prove this in detail for types only, and the case for terms is analogous. It suffices, using the Kripke-Joyal semantics of the topos $\widehat{\square}_+$, to fix $\Psi : \square_+$ and show that for all $r \in \mathbb{I}(\Psi)$, $n \in \mathbb{N}$, and $A, B \in (\!|\mathsf{Ty}_n|\!)(\Psi)$, if $\Psi \Vdash (\partial(r) \implies A = B)$ then $\Psi \Vdash A = B$.

We will write $\Psi.\xi_\varepsilon \xrightarrow{\widehat{\xi_\varepsilon}} \Psi$ for the equalizer of r, ε for each $\varepsilon \in \mathbf{2}$. Recalling that $(\!|\mathsf{Ty}_n|\!)(\Psi) = \mathsf{Ty}^n_{\mathcal{C}}(\langle\Psi\rangle)$, by the $\mathbf{K}_\partial$-separation of $\mathsf{Ty}^n_{\mathcal{C}}$ (see Definition B.8), to show that $\Psi \Vdash A = B$ it suffices to show that $\Psi.\xi_e \Vdash \widehat{\xi_\varepsilon}^* A = \widehat{\xi_\varepsilon}^* B$ for each $\varepsilon \in \mathbf{2}$.

Unraveling the Kripke-Joyal paperwork of our assumption, we know that for all $\Psi' \xrightarrow{\psi} \Psi$, if $\Psi' \Vdash \partial(r \circ \psi)$ then $\Psi' \Vdash \psi^* A = \psi^* B$. Instantiating this hypothesis with the equalizer $\Psi.\xi_e \xrightarrow{\widehat{\xi_\varepsilon}} \Psi$, it remains only to check that $\Psi.\xi_\varepsilon \Vdash (r \circ \hat{\xi_\varepsilon} = 0) \vee (r \circ \hat{\xi_\varepsilon} = 1)$. But we immediately have $\Psi.\xi_0 \Vdash r \circ \hat{\xi_0} = 0$ and $\Psi.\xi_1 \Vdash r \circ \hat{\xi_1} = 1$ by the property of the equalizer. ◀

▶ **Definition D.4.** *Given $A : (\!|\mathsf{Ty}_n|\!)$ and $\texttt{A} : \mathfrak{U}^\bullet_n A$, we say that $\texttt{A}$ is* elementwise separated *(or just* separated*) iff the following formula holds internally to $\widehat{\square}_+$:*

$$\forall r : \mathbb{I}, M : (\!|A|\!), M^\bullet_0, M^\bullet_1 : \texttt{A}^\circ M, (\partial(r) \implies M^\bullet_0 = M^\bullet_1) \implies M^\bullet_0 = M^\bullet_1$$

We write $\mathsf{Separated}(\texttt{A})$ *for the above formula.*

▶ **Definition D.5.** *Given $A : (\!|\mathsf{Ty}_n|\!)$ and $\texttt{A} : \mathfrak{U}^\bullet_n A$, we say that $\texttt{A}$ is* typewise separated *iff the following formula holds internally to $\widehat{\square}_+$:*

$$\forall r : \mathbb{I}, B : (\!|\mathsf{Ty}_n|\!), \texttt{B} : \mathfrak{U}^\bullet_n B.\, (\partial(r) \implies \texttt{A} = \texttt{B}) \implies \texttt{A} = \texttt{B}$$

We write $\mathsf{TypewiseSeparated}(\texttt{A})$ *for the above formula.*

▶ **Lemma D.6.** *Internally to $\widehat{\square}_+$, the following formula is true:*

$$\forall n : \mathbb{N}, A : (\!|\mathsf{Ty}_n|\!), \mathtt{A} : \mathfrak{U}^\bullet_n A.\ \mathsf{TypewiseSeparated}(\mathtt{A}) \wedge \mathsf{Separated}(\mathtt{A})$$

Proof. We begin by strong induction on n; then, we proceed by induction on $\mathtt{A}$.

1. *Case (dependent function type).* Fixing $\mathtt{C} : \mathfrak{U}^\bullet_n C$ and $\mathtt{D} : \mathfrak{F}_n[\mathtt{C}]^\bullet D$, we have to verify that $\mathtt{pi}(\mathtt{C}; \mathtt{D})$ is typewise and elementwise separated.
   a. *Typewise separation.* We need to show that for all $\mathtt{B} : \mathfrak{U}^\bullet_n B$, if $\partial(r) \implies \mathtt{pi}(\mathtt{C}; \mathtt{D}) = \mathtt{B}$, then $\mathtt{pi}(\mathtt{C}; \mathtt{D}) = \mathtt{B}$. First we observe that there exist $\mathtt{E}, \mathtt{F}$ such that $\mathtt{B} = \mathtt{pi}(\mathtt{E}; \mathtt{F})$. This follows by inversion on $\mathtt{B}$: if $\mathtt{B}$ is a $\mathtt{pi}$ our goal is immediate, otherwise we would have $\partial(r) \implies \bot$ and in $\widehat{\square}_+$ we have that $(\partial(r) \implies \bot) \implies \bot$ giving a contradiction. Therefore, we have some $\mathtt{E}$ and $\mathtt{F}$ such that $\partial(r) \implies \mathtt{pi}(\mathtt{C}; \mathtt{D}) = \mathtt{pi}(\mathtt{E}; \mathtt{F})$. By inversion, we have $\partial(r) \implies (\mathtt{C} = \mathtt{E} \wedge \mathtt{D} = \mathtt{F})$. We need to show that $\mathtt{C} = \mathtt{E}$ and $\mathtt{D} = \mathtt{F}$.
      i. Instantiating our induction hypothesis for the typewise separation of $\mathtt{C}$ with $\mathtt{E}$, we obtain $\mathtt{C} = \mathtt{E}$ from $\partial(r) \implies \mathtt{C} = \mathtt{E}$.
      ii. To see that $\mathtt{D} = \mathtt{F}$, we fix $M : (\!|C|\!)$ and $M^\bullet : \mathtt{C}^\circ M$, and show that $\mathtt{D}MM^\bullet = \mathtt{F}MM^\bullet$. Instantiating our induction hypothesis for the typewise separation of $\mathtt{D}MM^\bullet$, we obtain $\mathtt{D}MM^\bullet = \mathtt{F}MM^\bullet$ from $\partial(r) \implies \mathtt{D} = \mathtt{F}$.
   b. *Elementwise separation* Fixing $M : (\!|\mathbf{\Pi}(C, D)|\!)$ and $M_0^\bullet, M_1^\bullet : \mathtt{pi}(\mathtt{C}; \mathtt{D})^\circ M$ such that $\partial(r) \implies M_0^\bullet = N_1^\bullet$, we need to show that $M_0^\bullet = M_1^\bullet$. We fix $N : (\!|C|\!)$ and $N^\bullet : \mathtt{C}^\circ N$, to verify that $M_0^\bullet NN^\bullet = M_1^\bullet NN^\bullet$. Using our induction hypothesis for the elementwise separation of $\mathtt{D}NN^\bullet$, we obtain $M_0^\bullet NN^\bullet = N_1^\bullet N^\bullet$ from $\partial(r) \implies M_0^\bullet NN^\bullet = M_1^\bullet NN^\bullet$.
2. *Case (dependent pair type).* Fixing $\mathtt{C} : \mathfrak{U}^\bullet_n C$ and $\mathtt{D} : \mathfrak{F}_n[\mathtt{C}]^\bullet D$, we have to verify that $\mathtt{sg}(\mathtt{C}; \mathtt{D})$ is typewise and elementwise separated.
   a. *Typewise separation.* This case is identical to the case for dependent function types.
   b. *Elementwise separation.* Fixing $M : (\!|\mathbf{\Sigma}(C, D)|\!)$ and $M_0^\bullet, M_1^\bullet : \mathtt{sg}(\mathtt{C}; \mathtt{D})^\circ M$ such that $\partial(r) \implies M_0^\bullet = M_1^\bullet$, we need to show that $M_0^\bullet = M_1^\bullet$. It suffices to show that $\pi_1 M_0^\bullet = \pi_2 M_1^\bullet$ and $\pi_2 M_0^\bullet = \pi_2 M_1^\bullet$.
      i. From our induction hypothesis for the elementwise separation of $\mathtt{C}$, we obtain $\pi_1 M_0^\bullet = \pi_1 M_1^\bullet$ from $\partial(r) \implies M_0^\bullet = M_1^\bullet$.
      ii. From our induction hypothesis for the elementwise separation of $\mathtt{D}MM_0^\bullet$, we obtain $\pi_2 M_0^\bullet = \pi_2 M_1^\bullet$ from $\partial(r) \implies M_0^\bullet = M_1^\bullet$.
3. *Case (equality type).* Fixing $\mathtt{C} : \prod_i \mathfrak{U}^\bullet_n C_i$ and $M_0 : (\!|C_0|\!), M_0^\bullet : \mathtt{C0}^\circ M_0$ and $M_1 : (\!|C_1|\!), M_1^\bullet : \mathtt{C1}^\circ M_1$, we have to verify that $\mathtt{eq}(\mathtt{C}; M_0^\bullet, M_1^\bullet)$ is typewise and elementwise separated.
   a. *Typewise separation.* We need to show that for all $\mathtt{B} : \mathfrak{U}^\bullet_n B$, if $\partial(r) \implies \mathtt{eq}(\mathtt{C}; M_0^\bullet, M_1^\bullet) = \mathtt{B}$, then $\mathtt{eq}(\mathtt{C}; M_0^\bullet, M_1^\bullet) = \mathtt{B}$. By inversion, we observe that there exist $\mathtt{D} : \prod_i \mathfrak{U}^\bullet_n C_i$ and $N_0^\bullet : \mathtt{D0}^\circ M_0$ and $N_1^\bullet : \mathtt{D1}^\circ M_1$ such that $\partial(r) \implies \mathtt{B} = \mathtt{eq}(\mathtt{D}; N_0^\bullet, N_1^\bullet)$. By inversion we have $\partial(r) \implies (\mathtt{D} = \mathtt{C} \wedge \overrightarrow{M_\varepsilon^\bullet = N_\varepsilon^\bullet})$. We need to show that $\mathtt{D} = \mathtt{C}$ and $\overrightarrow{M_\varepsilon^\bullet = N_\varepsilon^\bullet}$.
      i. To see that $\mathtt{D} = \mathtt{C}$, we fix $i : \mathbb{I}$ and verify that $\mathtt{D}i = \mathtt{C}i$. Instantiating our induction hypothesis for the typewise separation of $\mathtt{D}i$, we obtain $\mathtt{D}i = \mathtt{C}i$ from $\partial(r) \implies \mathtt{D} = \mathtt{C}$.
      ii. To see that $M_\varepsilon^\bullet = N_\varepsilon^\bullet$, we instantiate our induction hypothesis for the elementwise separation of $\mathtt{D}\varepsilon$, obtaining $M_\varepsilon^\bullet = N_\varepsilon^\bullet$ from $\partial(r) \implies M_\varepsilon^\bullet = N_\varepsilon^\bullet$.
   b. *Elementwise separation.* Fixing $P : (\!|\mathbf{Eq}(i.C_i, M_0, M_1)|\!)$ and $P_0^\bullet, P_1^\bullet : \mathtt{eq}(\mathtt{C}; M_0^\bullet, M_1^\bullet)^\circ P$ such that $\partial(r) \implies P_0^\bullet = P_1^\bullet$, we need to show that $P_0^\bullet = P_1^\bullet$; fixing $i : \mathbb{I}$, we verify that $P_0^\bullet i = P_1^\bullet i$. Using our induction hypothesis for the elementwise separation of $\mathtt{C}i$, we obtain $P_0^\bullet i = P_1^\bullet i$ from $\partial(r) \implies P_0^\bullet i = P_1^\bullet$.
4. *Case (boolean type).* We need to show that $\mathtt{bool}$ is typewise and elementwise separated.

a. *Typewise separation.* We need to show that for all $\mathtt{B} : \mathfrak{U}^\bullet_n B$, if $\partial(r) \implies \mathtt{bool} = \mathtt{B}$, then $\mathtt{bool} = \mathtt{B}$. But this is immediate by considering the restriction maps for $\mathtt{bool}$.

b. *Elementwise separation.* Fixing $M : (\!|\mathbf{bool}|\!)$ and $M_0^\bullet, M_1^\bullet : \mathtt{bool}^\circ M$ such that $\partial(r) \implies M_0^\bullet = M_1^\bullet$, we need to show that $M_0^\bullet = M_1^\bullet$. We obtain our goal by case on $M_0^\bullet, M_1^\bullet$, observing that the cross-cases $\partial(r) \implies \mathbf{inl}(\ldots) = \mathbf{inr}(\ldots)$ and $\partial(r) \implies \mathbf{inr}(\ldots) = \mathbf{inl}(\ldots)$ are impossible.

5. *Case (universe).* We need to show that $\mathtt{univ}_m$ is typewise and elementwise separated for $m < n$.

a. *Typewise separation.* We need to show that for all $\mathtt{B} : \mathfrak{U}^\bullet_n B$, if $\partial(r) \implies \mathtt{univ}_m = \mathtt{B}$, then $\mathtt{univ}_m = \mathtt{B}$. But this is immediate by considering the restriction maps for $\mathfrak{U}_n$.

b. *Elementwise separation.* Elementwise separation of $\mathtt{univ}_m$ follows from the typewise separation part of the outer induction hypothesis at $m < n$. ◀

D.2 The universe type and its decoding

Next, we define a hierarchy of closed universes à la Tarski $\overline{\mathbf{U}_n} \in \mathsf{Ty}_{\mathcal{C}^\bullet}(\overline{\Gamma})$. Note that these are *not* universes à la Russell in the sense of Definition B.17. For the syntactic part, take $\mathbf{U}_n \in \mathsf{Ty}_{\mathcal{C}}(\Gamma)$ itself; then, we define the computability family using $\mathfrak{U}^\bullet_n$:

$$\begin{aligned}
\mathbf{U}^\bullet_n &: \textstyle\prod_{\gamma:(\!|\Gamma|\!)} \prod_{\gamma^\bullet:\Gamma^\bullet\gamma} (\!|\gamma^* \mathbf{U}_n|\!) \to \mathcal{V}_{n+1} \\
\ldots &: \textstyle\prod_{\gamma:(\!|\Gamma|\!)} \prod_{\gamma^\bullet:\Gamma^\bullet\gamma} (\!|\mathsf{Ty}_n|\!) \to \mathcal{V}_{n+1} \\
\mathbf{U}^\bullet_n \gamma\gamma^\bullet A &= \mathfrak{U}^\bullet_n A
\end{aligned}$$

We equip each type with regular coercion and homogeneous composition structure for these universes, by recursion on the type codes. For readability, we leave syntactic arguments like $A, M, \ldots$ implicit.

$$\frac{r, r' : \mathbb{I} \quad \mathtt{A} : \prod_{i:\mathbb{I}} \mathfrak{U}^\bullet_n A_i \quad M^\bullet : \mathtt{A}r^\circ M}{[i.\mathtt{A}i] \downarrow^r_{r'} M^\bullet : \mathtt{A}r'^\circ\left(\mathbf{coe}^{r \rightsquigarrow r'}_{i.A_i}\, M\right)}$$

$$\frac{r, r' : \mathbb{I} \quad \mathtt{A} : \mathfrak{U}^\bullet_n A \quad M^\bullet : \mathtt{A}^\circ M \quad \overrightarrow{N^\bullet_\varepsilon : \prod_{i:\mathbb{I}} (s = \varepsilon) \to \{x : \mathtt{A}^\bullet N_i \mid i = r \implies x = M^\bullet\}}}{\mathtt{A} \downarrow^r_{r'} M^\bullet\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.N^\bullet_\varepsilon i}] : \mathtt{A}^\circ \left(\mathbf{hcom}^{r \rightsquigarrow r'}_A\, M\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.N_i}]\right)}$$

$$\begin{aligned}
[i.\mathtt{bool}] \downarrow^r_{r'} M^\bullet &= M^\bullet \\
[i.\mathtt{univ}_n] \downarrow^r_{r'} \mathtt{A} &= \mathtt{A} \\
[i.\mathtt{pi}(\mathtt{A}; \mathtt{B})] \downarrow^r_{r'} M^\bullet &= \lambda N^\bullet.\, [i.\mathtt{B}([i.\mathtt{A}] \downarrow^{r'}_i N^\bullet)] \downarrow^r_{r'} M^\bullet\left([i.\mathtt{A}] \downarrow^{r'}_r N^\bullet\right) \\
[i.\mathtt{sg}(\mathtt{A}; \mathtt{B})] \downarrow^r_{r'} (M_0^\bullet, M_1^\bullet) &= \left([i.\mathtt{A}] \downarrow^r_{r'} M_0^\bullet, [i.\mathtt{B}([i.\mathtt{A}] \downarrow^r_i M_0^\bullet)] \downarrow^r_{r'} M_1^\bullet\right) \\
[i.\mathtt{eq}(\mathtt{A}; N_0^\bullet, N_1^\bullet)] \downarrow^r_{r'} M^\bullet &= \lambda k.\, [i.\mathtt{A}k] \downarrow^r_{r'} M^\bullet k\ [k \text{ with } \overrightarrow{\varepsilon \hookrightarrow _.N^\bullet_\varepsilon}]
\end{aligned}$$

$$\frac{\widetilde{M^\bullet} = [i.\mathtt{A}_i] \downarrow^r_{r'} M^\bullet \qquad \widetilde{N\varepsilon^\bullet} = \lambda i.\, [i.\mathtt{A}_i] \downarrow^i_{r'} N^\bullet_i}{[i.\mathtt{A}_i] \downarrow^r_{r'} M^\bullet\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.N^\bullet_\varepsilon i}] = \mathtt{A}_{r'} \downarrow^r_{r'} \widetilde{M^\bullet}\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.N^\bullet_\varepsilon i}]}$$

$$\mathtt{univ}_n \downarrow^r_{r'} \mathtt{bool}\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow \_.\mathtt{bool}}] = \mathtt{bool}$$

$$\mathtt{univ}_n \downarrow^r_{r'} \mathtt{univ}_k\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow \_.\mathtt{univ}_k}] = \mathtt{univ}_k$$

$$\frac{\widetilde{\mathtt{A}} = \lambda i.\, \mathtt{univ}_n \downarrow^r_i \mathtt{A}\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.\mathtt{A}'i}] \qquad \widetilde{\mathtt{B}} = \lambda N^\bullet.\, \mathtt{univ}_n \downarrow^r_{r'} \mathtt{B}\big([i.\widetilde{\mathtt{A}}i] \downarrow^{r'}_r N^\bullet\big)\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.\mathtt{B}'i\big([i.\widetilde{\mathtt{A}}i] \downarrow^{r'}_i N^\bullet\big)}]}{\mathtt{univ}_n \downarrow^r_{r'} \mathtt{pi}(\mathtt{A};\mathtt{B})\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.\mathtt{pi}(\mathtt{A}'i;\mathtt{B}'i)}] = \mathtt{pi}\big(\widetilde{\mathtt{A}}r';\widetilde{\mathtt{B}}\big)}$$

$$\frac{\widetilde{\mathtt{A}} = \lambda i.\, \mathtt{univ}_n \downarrow^r_i \mathtt{A}\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.\mathtt{A}'i}] \qquad \widetilde{\mathtt{B}} = \lambda N^\bullet.\, \mathtt{univ}_n \downarrow^r_{r'} \mathtt{B}\big([i.\widetilde{\mathtt{A}}i] \downarrow^{r'}_r N^\bullet\big)\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.\mathtt{B}'i\big([i.\widetilde{\mathtt{A}}i] \downarrow^{r'}_i N^\bullet\big)}]}{\mathtt{univ}_n \downarrow^r_{r'} \mathtt{sg}(\mathtt{A};\mathtt{B})\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.\mathtt{sg}(\mathtt{A}'i;\mathtt{B}'i)}] = \mathtt{sg}\big(\widetilde{\mathtt{A}}r';\widetilde{\mathtt{B}}\big)}$$

$$\frac{\widetilde{\mathtt{A}} = \lambda j, i.\, \mathtt{univ}_n \downarrow^r_j \mathtt{A}i\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.\mathtt{A}'ji}] \qquad \widetilde{M} = [j.\widetilde{\mathtt{A}}jr] \downarrow^r_{r'} M^\bullet\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.M'^\bullet j}] \qquad \widetilde{N} = [j.\widetilde{\mathtt{A}}jr'] \downarrow^r_{r'} N^\bullet\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.N'^\bullet j}]}{\mathtt{univ}_n \downarrow^r_{r'} \mathtt{eq}(\mathtt{A}; M^\bullet, N^\bullet)\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.\mathtt{eq}(\mathtt{A}'i; M'^\bullet i, N'^\bullet i)}] = \mathtt{eq}\Big(\widetilde{\mathtt{A}}r'; \widetilde{M^\bullet}, \widetilde{N^\bullet}\Big)}$$

$$\mathtt{bool} \downarrow^r_{r'} \mathbf{inl}(\mathbf{refl})\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow \_.\mathbf{inl}(\mathbf{refl})}] = \mathbf{inl}(\mathbf{refl})$$

$$\mathtt{bool} \downarrow^r_{r'} \mathbf{inr}(\mathbf{refl})\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow \_.\mathbf{inr}(\mathbf{refl})}] = \mathbf{inr}(\mathbf{refl})$$

$$\mathtt{pi}(\mathtt{A};\mathtt{B}) \downarrow^r_{r'} M^\bullet\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.M'^\bullet i}] = \lambda N^\bullet.\, \mathtt{B}N^\bullet \downarrow^r_{r'} M^\bullet N^\bullet\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.M'^\bullet iNN^\bullet}]$$

$$\frac{\widetilde{M^\bullet} = \lambda j.\, \mathtt{A} \downarrow^r_j M^\bullet\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.M'^\bullet i}] \qquad \widetilde{N^\bullet} = [i.\mathtt{B}\big(\widetilde{M^\bullet} i\big)] \downarrow^r_{r'} N^\bullet\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow N'^\bullet i}]}{\mathtt{sg}(\mathtt{A};\mathtt{B}) \downarrow^r_{r'} (M^\bullet, N^\bullet)\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.(M'^\bullet i, N'^\bullet i)}] = \big(\widetilde{M^\bullet}r', \widetilde{N^\bullet}\big)}$$

$$\mathtt{eq}(\mathtt{A}; N_0^\bullet, N_1^\bullet) \downarrow^r_{r'} M^\bullet\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.M'^\bullet i}] = \lambda j.\, \mathtt{A}j \downarrow^r_{r'} M^\bullet j\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow i.M'^\bullet ij}]$$

Next, we show that every element $\overline{A} \equiv (A, \mathtt{A}) \in \mathsf{El}_{\mathcal{C}^\bullet}(\overline{\Gamma} \vdash \overline{\mathbf{U}_n})$ determines a type $\mathbf{T}[\overline{A}] \in \mathsf{Ty}^n_{\mathcal{C}^\bullet}(\overline{\Gamma})$. For the syntactic part, choose $A \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \mathbf{U}_n)$ itself (which is possible because $\mathbf{U}_n$ is a universe à la Russell in $\mathcal{C}$). The computability family is given as follows:

$$\mathbf{T}[\overline{A}]^\bullet \gamma\gamma^\bullet = (\mathtt{A}\gamma\gamma^\bullet)^\circ$$

D.3 The *closed-universe* computability cwf

Now, we are equipped to build a *new* cwf $\mathcal{C}^\star$, which we will show to be a model of XTT in Appendix D.4. Let the underlying category of $\mathcal{C}^\star$ be the same as $\mathcal{C}^\bullet$'s; we will choose new presheaves of types and elements, however.

$$\begin{aligned} \mathsf{Ty}^n_{\mathcal{C}^\star}(\overline{\Gamma}) &= \mathsf{El}_{\mathcal{C}^\bullet}(\overline{\Gamma} \vdash \overline{\mathbf{U}_n}) \\ \mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma} \vdash \overline{A}) &= \mathsf{El}_{\mathcal{C}^\bullet}(\overline{\Gamma} \vdash \mathbf{T}[\overline{A}]) \end{aligned}$$

When $\overline{A} \in \mathsf{Ty}^k_{\mathcal{C}^\star}(\overline{\Gamma})$ and $k \leq l$, we exhibit the level restriction by taking $\Uparrow^l_k A$ for the syntactic part, and retaining $A^\bullet$ for the semantic part, a move justified by the fact that the type codes are invariant under lifting (Lemma D.1).

Given $\overline{\Gamma} : \mathcal{C}^\star$ and $\overline{A} \in \mathsf{Ty}^n_{\mathcal{C}^\star}(\overline{\Gamma})$, we need to exhibit the context comprehension $\overline{\Gamma}.\overline{A} : \mathcal{C}^\star$ with the appropriate projection map $\overline{\mathbf{p}}$ and variable term $\overline{\mathbf{q}}$. We choose the already-existing context comprehension $\overline{\Gamma}.\mathbf{T}[\overline{A}]$ inherited from $\mathcal{C}^\bullet$; the projection map and variable term are likewise inherited. From Lemma D.1 we also immediately obtain $\mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma} \vdash \overline{A}) = \mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma} \vdash \Uparrow^l_k \overline{A})$ and $\overline{\Gamma}.\overline{A} = \overline{\Gamma}.\Uparrow^l_k \overline{A}$.

The projection π_{syn} lifts from $\mathcal{C}^\bullet$ to a fibration $\mathcal{C}^\star \xrightarrow{\pi_{\mathsf{syn}}} \mathcal{C}$, because the underlying categories of $\mathcal{C}^\bullet$ and $\mathcal{C}^\star$ are identical. A cubical structure for $\mathcal{C}^\star$ is obtained from the composite $\mathcal{C}^\star \xrightarrow{\pi_{\mathsf{syn}}} \mathcal{C} \xrightarrow{\mathfrak{u}} \square_+$.

D.4 A logical families model of XTT

In this section, we argue that the cwf $\mathcal{C}^\star$ has the structure of a model of XTT.

▶ **Construction D.7.** In preparation, we first will observe that *internally to* $\widehat{\square}_+$, we have the following operations for any $\overline{\Gamma} : \mathcal{C}^\star$ and fresh i:

$$(-.-) : (\!|\Gamma|\!) \times \mathbb{I} \to (\!|\hat{\imath}^*\Gamma|\!) \tag{1}$$

$$(-.-) : \textstyle\prod_{\gamma:(\!|\Gamma|\!)} \prod_{r:\mathbb{I}} \Gamma^\bullet\gamma \to (\hat{\imath}^*\Gamma)^\bullet(\gamma.r) \tag{2}$$

1. Fix Ψ and $\langle\Psi\rangle \xrightarrow{\gamma} \Gamma$ and $\Psi \mid r\ dim$, defining:

$$\langle\Psi\rangle \xrightarrow{\langle r/i\rangle} \langle\Psi, i\rangle \equiv \hat{\imath}^*\langle\Psi\rangle \xrightarrow{\hat{\imath}^+\gamma} \hat{\imath}^*\Gamma$$

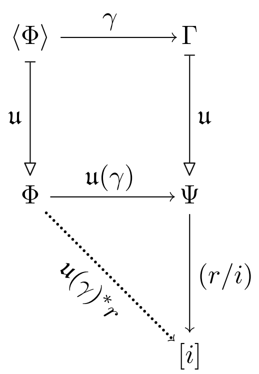
   Naturality is just the associativity of composition.

2. Fix Ψ and $\langle\Psi\rangle \xrightarrow{\gamma} \Gamma$ and $\gamma^\bullet \in \Gamma^\bullet_\Psi\gamma$ and $\Psi \mid r\ dim$. We need to construct some element of $(\hat{\imath}^*\Gamma)^\bullet_\Psi(\gamma.r)$. Observing that $(\gamma, \gamma^\bullet)$ constitute a map $\overline{\langle\Psi\rangle} \xrightarrow{\overline{\gamma}} \overline{\Gamma}$ in $\mathcal{C}^\star$, we see that our goal is in fact to transform $\overline{\gamma}$ into a map $\overline{\langle\Psi\rangle} \longrightarrow \hat{\imath}^*\overline{\Gamma}$ which lies over $\gamma.r$. This we obtain in the same way as before, this time using the composite fibration $\mathcal{C}^\star \xrightarrow{\pi_{\mathsf{syn}}} \mathcal{C} \xrightarrow{\mathfrak{u}} \square_+$:

$$\overline{\langle\Psi\rangle} \xrightarrow{\overline{\langle r/i\rangle}} \overline{\langle\Psi, i\rangle} \equiv \hat{\imath}^*\overline{\langle\Psi\rangle} \xrightarrow{\hat{\imath}^+\overline{\gamma}} \hat{\imath}^*\overline{\Gamma}$$

   By analogy, we write this map as $\gamma^\bullet.r$. ◀

▶ **Construction D.8.** When $\Gamma : \mathcal{C}$ lies over Ψ and $\Psi \mid r\ dim$, we obtain a natural transformation $(\!|\Gamma|\!) \to \mathbb{I}$; fixing $\langle\Phi\rangle \xrightarrow{\gamma} \Gamma$, we obtain:



We will write $\gamma[r] : \mathbb{I}$ given $\gamma : (\!|\Gamma|\!)$ and $\Psi \mid r\ dim$ when working internally. ◀

▶ **Proposition D.9.** *If* $\eta : (\!|\Gamma.r = s|\!)$*, then* $\eta[r] = \eta[s] : \mathbb{I}$.

▶ **Lemma D.10.** *$\mathcal{C}^\star$ has regular coercion structure in the sense of Definition B.9.*

Proof. Fixing $\overline{A} \in \mathsf{Ty}_{\mathcal{C}^\star}(\hat{\imath}^*\overline{\Gamma})$ over Ψ, i and dimensions $\Psi \mid r, r'$ *dim* and an element $\overline{M} \in \mathsf{El}_{\mathcal{C}}(\overline{\Gamma} \vdash (r/i)^\ddagger A)$, we must construct an element $\mathbf{coe}_{i.\overline{A}}^{r \rightsquigarrow r'} \overline{M}$ which satisfies the adjacency, regularity and naturality equations. Taking $\mathbf{coe}_{i.\overline{A}}^{r \rightsquigarrow r'} M$ for the syntactic part, we construct its realizer as follows:

$$\left(\mathbf{coe}_{i.\overline{A}}^{r \rightsquigarrow r'} \overline{M}\right)^\bullet \gamma\gamma^\bullet = [j.A^\bullet(\gamma.j)(\gamma^\bullet.j)] \downarrow_{\gamma[r']}^{\gamma[r]} (M^\bullet\gamma\gamma^\bullet)$$

- *Adjacency.* If $r = r'$ then $\mathbf{coe}_{i.\overline{A}}^{r \rightsquigarrow r'} \overline{M} = \overline{M}$. This holds already for the syntactic part, so it remains to see that our realizer preserves it; this is proved by induction on the graph of the realizer for coercion in Appendix D.2.
- *Regularity.* If $\overline{A} = \hat{\imath}^\ddagger \overline{A'}$ for some $\overline{A'} \in \mathsf{Ty}_{\mathcal{C}^\star}(\overline{\Gamma})$, then $\mathbf{coe}_{i.\overline{A}}^{r \rightsquigarrow r'} \overline{M} = \overline{M}$. As above, we need only observe that the realizer exhibits regularity, which is evident by inspecting each of the clauses in Appendix D.2.
- *Naturality.* For $\overline{\Delta} \xrightarrow{\overline{\gamma}} \overline{\Gamma}$ we have $\overline{\gamma}^*\mathbf{coe}_{i.\overline{A}}^{r \rightsquigarrow r'} \overline{M} = \mathbf{coe}_{i.(\hat{\imath}+\overline{\gamma})^*\overline{A}}^{\mathsf{u}(\overline{\gamma})^*r \rightsquigarrow \mathsf{u}(\overline{\gamma})^*r'} \overline{\gamma}^*\overline{M}$. As above, this holds already of the syntactic part, so we need only to verify for each $\delta : (\!|\Delta|\!), \delta^\bullet : \Delta^\bullet\delta$ the following:

$$[j.A^\bullet((\gamma^*\delta).j)((\gamma^*\delta^\bullet).j)]\downarrow_{(\gamma^*\delta)[r']}^{(\gamma^*\delta)[r]} M^\bullet(\gamma^*\delta)(\gamma^*\delta^\bullet) = [j.((\hat{\imath}^+\gamma)^*A^\bullet)(\delta.j)(\delta^\bullet.j)]\downarrow_{\delta[\overline{\gamma}^*r']}^{\delta[\overline{\gamma}^*r]} (\overline{\gamma}^*M^\bullet)\delta\delta^\bullet$$

The above follows from the naturality of Construction D.8. ◀

▶ **Lemma D.11.** *$\mathcal{C}^\star$ has regular homogeneous composition structure in the sense of Definition B.10.*

Proof. Fix a type $\overline{A} \in \mathsf{Ty}_{\mathcal{C}^\star}(\Gamma)$, dimensions $r, r', s \in \mathsf{Dim}_{\mathcal{C}}(\overline{\Gamma})$, a cap $\overline{M} \in \mathsf{El}_{\mathcal{C}}(\overline{\Gamma} \vdash \overline{A})$ and a tube $\overrightarrow{\overline{N_\varepsilon} \in \mathsf{El}_{\mathcal{C}}(\hat{\jmath}^*(\overline{\Gamma}.s = \varepsilon) \vdash \hat{\jmath}^\ddagger \widehat{s = \varepsilon}^*\overline{A})}$ for fresh j such that $\overrightarrow{(r/j)^\ddagger\overline{N_\varepsilon} = \overline{M}}$. Choosing

$$\begin{aligned}&\left(\mathbf{hcom}_{\overline{A}}^{r \rightsquigarrow r'} \overline{M}\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.\overline{N_\varepsilon}}]\right)^\bullet \gamma\gamma^\bullet \\ &= A^\bullet\gamma\gamma^\bullet \downarrow_{\gamma[r']}^{\gamma[r]} M^\bullet\gamma\gamma^\bullet\ [s \text{ with } \overrightarrow{\varepsilon \hookrightarrow j.N_\varepsilon^\bullet(\gamma.s = \varepsilon.j)(\gamma^\bullet.j)}]\end{aligned}$$

The adjacency and regularity conditions hold by induction on the graph of the realizer defined in Appendix D.2. The naturality condition lifts directly from $\mathcal{C}$ as in Lemma D.10. ◀

▶ **Lemma D.12.** *$\mathcal{C}^\star$ has the boolean type, in the sense of Definition B.12.*

Proof.

- *Formation.* To exhibit $\overline{\mathbf{bool}} \in \mathsf{Ty}_{\mathcal{C}^\star}^n(\overline{\Gamma})$, we choose **bool** itself for the syntactic part, and for its realizer we choose $\mathbf{bool}^\bullet\gamma\gamma^\bullet = \texttt{bool}$.
- *Introduction.* Choosing **true** and **false** for the syntactic parts, we exhibit realizers as follows:

$$\mathbf{true}^\bullet\gamma\gamma^\bullet = \mathbf{inl}(\mathbf{refl})$$
$$\mathbf{false}^\bullet\gamma\gamma^\bullet = \mathbf{inr}(\mathbf{refl})$$

- *Elimination.* Fixing $\overline{C} \in \mathsf{Ty}_{\mathcal{C}^\star}^n(\overline{\Gamma}.\overline{\mathbf{bool}})$ and $\overline{M} \in \mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma} \vdash \overline{\mathbf{bool}})$ and $\overline{N_0} \in \mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma} \vdash \langle \mathbf{id}, \overline{\mathbf{true}}\rangle^* C)$ and $\overline{N_1} \in \mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma} \vdash \langle \mathbf{id}, \overline{\mathbf{false}}\rangle^* C)$, we choose $\mathbf{if}_C(M, N_0, N_1) \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash \langle \mathbf{id}, M\rangle^*)C$ for the syntactic part, exhibiting its realizer as follows:

$$\mathbf{if}_{\overline{C}}(\overline{M}, \overline{N_0}, \overline{N_1})^\bullet \gamma\gamma^\bullet = \begin{cases} N_0^\bullet\gamma\gamma^\bullet & \text{if } M^\bullet\gamma\gamma^\bullet = \mathbf{inl}(\ldots) \\ N_1^\bullet\gamma\gamma^\bullet & \text{if } M^\bullet\gamma\gamma^\bullet = \mathbf{inr}(\ldots)\end{cases}$$

- *Computation, naturality.* The satisfaction of the equational conditions is immediate. ◀

▶ **Lemma D.13.** *$\mathcal{C}^\star$ has dependent function types in the sense of Definition B.13.*

Proof.

- *Formation.* Fixing $\overline{A} \in \mathsf{Ty}^n_{\mathcal{C}^\star}(\overline{\Gamma})$ and a family $\overline{B} \in \mathsf{Ty}^n_{\mathcal{C}^\star}(\overline{\Gamma}.\overline{A})$, we must exhibit a type $\mathbf{\Pi}(\overline{A}, \overline{B}) \in \mathsf{Ty}^n_{\mathcal{C}^\star}(\overline{\Gamma})$. Choosing $\mathbf{\Pi}(A, B)$ for the syntactic part, we must exhibit a realizer for the type:

$$\mathbf{\Pi}(\overline{A}, \overline{B})^\bullet \gamma\gamma^\bullet = \mathtt{pi}(A^\bullet\gamma\gamma^\bullet; \lambda N, N^\bullet.\, B^\bullet\langle\gamma, N\rangle(\gamma^\bullet, N^\bullet))$$

- *Introduction.* Fixing $\overline{M} \in \mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma}.\overline{A} \vdash \overline{B})$, we must exhibit an element $\mathbf{lam}(\overline{M}) \in \mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma} \vdash \mathbf{\Pi}(\overline{A}, \overline{B}))$. Choosing $\mathbf{lam}(M)$ for the syntactic part, we code its realizer as follows:

$$\mathbf{lam}(\overline{M})^\bullet \gamma\gamma^\bullet N N^\bullet = M^\bullet\langle\gamma, N\rangle(\gamma^\bullet, N^\bullet)$$

- *Elimination.* Fixing $\overline{M} \in \mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma} \vdash \mathbf{\Pi}(\overline{A}, \overline{B}))$ and $\overline{N} \in \mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma} \vdash \overline{A})$, we need an element $\mathbf{app}(\overline{A}, \overline{B}, \overline{M}, \overline{N}) \in \mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma} \vdash \langle\mathbf{id}, \overline{N}\rangle^*\overline{B})$. Choosing $\mathbf{app}(A, B, M, N)$ for the syntactic part, we code its realizer:

$$\mathbf{app}(\overline{A}, \overline{B}, \overline{M}, \overline{N})^\bullet \gamma\gamma^\bullet = M^\bullet\gamma\gamma^\bullet N N^\bullet$$

- *Computation, unicity, naturality.* These equations follow immediately from the fact that they hold of $\mathcal{C}$, and the corresponding properties of the dependent function types in $\widehat{\square}_+$.
- *Coercion.* Supposing $\overline{\Gamma} \overset{u}{\longmapsto} \Psi, i$ and $\Psi \mid r, r'$ *dim* and $\overline{M} \in \mathsf{El}_{\mathcal{C}^\star}((r/i)^*\overline{\Gamma} \vdash (r/i)^\ddagger\mathbf{\Pi}(\overline{A}, \overline{B}))$, we need to verify the following equation:

$$\mathbf{coe}^{r \rightsquigarrow r'}_{i.\mathbf{\Pi}(\overline{A},\overline{B})}\,\overline{M} = \mathbf{lam}(\mathbf{coe}^{r \rightsquigarrow r'}_{i.\langle\mathbf{id}, \mathbf{coe}^{r' \rightsquigarrow i}_{i.\mathbf{p}^*\overline{A}}\,\mathbf{q}\rangle^*\overline{B}}\,\mathbf{app}(\mathbf{p}^*(r/i)^\ddagger\overline{A}, \langle\mathbf{p}\circ\mathbf{p}, \mathbf{q}\rangle^*(r/i)^\ddagger\overline{B}, \mathbf{p}^*\overline{M}, \mathbf{coe}^{r' \rightsquigarrow r}_{i.\mathbf{p}^*\overline{A}}\,\mathbf{q}))$$

  As this holds already of the syntactic part, we need only verify that it holds of the realizers (determined in Lemma D.10); but this is immediate by the definition of realizer for coercion at `pi(A;B)` in Appendix D.2. ◀

▶ **Lemma D.14.** *$\mathcal{C}^\star$ has dependent pair types in the sense of Definition B.14.*

Proof.

- *Formation.* Fixing a type $\overline{A} \in \mathsf{Ty}^n_{\mathcal{C}^\star}(\overline{\Gamma})$ and a family $\overline{B} \in \mathsf{Ty}^n_{\mathcal{C}^\star}(\overline{\Gamma}.\overline{A})$, we choose $\mathbf{\Sigma}(A, B)$ for the syntactic part, and exhibit its realizer as follows:

$$\mathbf{\Sigma}(\overline{A}, \overline{B})^\bullet \gamma\gamma^\bullet = \mathtt{sg}(A^\bullet\gamma\gamma^\bullet; \lambda N, N^\bullet.\, B^\bullet\langle\gamma, N\rangle(\gamma^\bullet, N^\bullet))$$

- *Introduction.* Fixing $\overline{M} \in \mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma} \vdash \overline{A})$ and $\overline{N} \in \mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma} \vdash \langle\mathbf{id}, \overline{M}\rangle^*\overline{B})$, we choose $\mathbf{pair}(M, N)$ for the syntactic part, coding its realizer as follows:

$$\mathbf{pair}(\overline{M}, \overline{N})^\bullet \gamma\gamma^\bullet = (M^\bullet\gamma\gamma^\bullet, N^\bullet\gamma\gamma^\bullet)$$

- *Elimination.* Given $\overline{M} \in \mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma} \vdash \mathbf{\Sigma}(\overline{A}, \overline{B}))$, we choose $\mathbf{fst}(A, B, M) \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash A)$ and $\mathbf{snd}(A, B, M)$ for the syntactic parts of the elimination forms, and give their realizers as follows:

$$\begin{aligned} \mathbf{fst}(\overline{A}, \overline{B}, \overline{M})^\bullet \gamma\gamma^\bullet &= \pi_1(M^\bullet\gamma\gamma^\bullet) \\ \mathbf{snd}(\overline{A}, \overline{B}, \overline{M})^\bullet \gamma\gamma^\bullet &= \pi_2(M^\bullet\gamma\gamma^\bullet) \end{aligned}$$

- *Computation, unicity, naturality.* These are all immediate from the fact that they hold in $\mathcal{C}$, and the fact that analogous principles hold for the dependent pair types of $\widehat{\square}_+$.
- *Coercion.* Analogous to Lemma D.13. ◀

▶ **Lemma D.15.** *$\mathcal{C}^\star$ has dependent path types in the sense of Definition B.15.*

Proof. Here, we make use of Constructions D.7 and D.8.

- *Formation.* Fixing $\overline{A} \in \mathsf{Ty}^n_{\mathcal{C}^\star}(\hat{\imath}^*\overline{\Gamma})$ over Ψ, i and elements $\overrightarrow{\overline{N_\varepsilon} \in \mathsf{El}_{\mathcal{C}}(\overline{\Gamma} \vdash (\varepsilon/i)^\ddagger\overline{A})}$, we choose $\mathbf{Eq}(i.A, N_0, N_1)$ for the syntactic part, coding its realizer as follows:

$$\mathbf{Eq}(i.\overline{A}, \overline{N_0}, \overline{N_1})^\bullet\gamma\gamma^\bullet = \mathtt{eq}(\lambda j.\, A^\bullet(\gamma.j)(\gamma^\bullet.j); N_0^\bullet\gamma\gamma^\bullet, N_1^\bullet\gamma\gamma^\bullet)$$

- *Introduction.* Given $\overline{M} \in \mathsf{El}_{\mathcal{C}^\star}(\hat{\imath}^*\overline{\Gamma} \vdash \overline{A})$, we choose $\mathbf{plam}(i.M)$ for the syntactic part, and exhibit its realizer as follows:

$$\mathbf{plam}(i.M)^\bullet\gamma\gamma^\bullet = \lambda j.\, M^\bullet(\gamma.j)(\gamma^\bullet.j)$$

- *Elimination.* Fixing $\overline{M} \in \mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma} \vdash \mathbf{Path}(i.\overline{A}, \overline{N_0}, \overline{N_1}))$ and $\Psi \mid r$ *dim*, we choose $\mathbf{papp}(i.A, M, r)$ for the syntactic part; its realizer is analogous:

$$\mathbf{papp}(i.\overline{A}, \overline{M}, r)^\bullet\gamma\gamma^\bullet = M^\bullet\gamma\gamma^\bullet(\gamma[r])$$

- *Computation, boundary, unicity and naturality.* Immediate.
- *Coercion.* Analogous to Lemma D.13. ◀

▶ **Lemma D.16.** *$\mathcal{C}^\star$ has universes à la Russell in the sense of Definition B.17.*

Proof. Fixing $\overline{\Gamma} : \mathcal{C}^\star$ and levels $k < l$, we need a type $\overline{\mathbf{U}_k} \in \mathsf{Ty}^l_{\mathcal{C}^\star}(\overline{\Gamma})$ such that $\mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma} \vdash \overline{\mathbf{U}_k}) = \mathsf{Ty}^k_{\mathcal{C}^\star}(\overline{\Gamma})$. For the syntactic part, we simply choose $\mathbf{U}_k$; for its realizer:

$$\mathbf{U}_k^\bullet\gamma\gamma^\bullet = \mathtt{univ}_k$$

To see that the condition is met, we first observe that $\mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma} \vdash \overline{\mathbf{U}_k}) = \mathsf{El}_{\mathcal{C}^\bullet}(\overline{\Gamma} \vdash \mathbf{T}[\overline{\mathbf{U}_k}])$; and moreover, $\mathsf{Ty}^k_{\mathcal{C}^\star}(\overline{\Gamma}) = \mathsf{El}_{\mathcal{C}^\bullet}(\overline{\Gamma} \vdash \overline{\mathbf{U}_k})$. Therefore, it suffices to show that $\mathbf{T}[\overline{\mathbf{U}_k}] = \overline{\mathbf{U}_k}$.

$$\begin{aligned}\mathbf{T}[\overline{\mathbf{U}_k}] &= \left(\mathbf{U}_k, \lambda\gamma\gamma^\bullet.(\mathbf{U}_k^\bullet\gamma\gamma^\bullet)^\circ\right) \\ &= (\mathbf{U}_k, \lambda\gamma\gamma^\bullet.\mathtt{univ}_k^\circ) \\ &= (\mathbf{U}_k, \lambda\gamma\gamma^\bullet.\mathfrak{U}_k^\bullet) \\ &= \overline{\mathbf{U}_k}\end{aligned}$$
◀

▶ **Lemma D.17.** *$\mathcal{C}^\star$ has boundary separation in the sense of Definition B.8.*

Proof. Because $\mathcal{C}$ has boundary separation, we need only to see that this property lifts to the realizers. Therefore, it suffices to show the following:

- *Types.* For all $\overline{A}, \overline{B} \in \mathsf{Ty}_{\mathcal{C}^\star}(\overline{\Gamma})$ and $r \in \mathsf{Dim}_{\mathcal{C}^\star}(\overline{\Gamma})$, we must verify the following implication:

$$(\forall\gamma\gamma^\bullet.\, \overrightarrow{A^\bullet(\gamma.\gamma^*r = \varepsilon)\gamma^\bullet = B^\bullet(\gamma.\gamma^*r = \varepsilon)\gamma^\bullet}) \implies \forall\gamma\gamma^\bullet.\, A^\bullet\gamma\gamma^\bullet = B^\bullet\gamma\gamma^\bullet$$

Fixing γ and $\gamma^\bullet$, it suffices to show:

$$\overrightarrow{A^\bullet(\gamma.\gamma^*r = \varepsilon)\gamma^\bullet = B^\bullet(\gamma.\gamma^*r = \varepsilon)\gamma^\bullet} \implies A^\bullet\gamma\gamma^\bullet = B^\bullet\gamma\gamma^\bullet$$

But this is equivalent to the following, which is obtained from the typewise separation of $A^\bullet\gamma\gamma^\bullet$, a consequence of Lemma D.6:

$$(\partial(\gamma^*r) \implies A^\bullet\gamma\gamma^\bullet = B^\bullet\gamma\gamma^\bullet) \implies A^\bullet\gamma\gamma^\bullet = B^\bullet\gamma\gamma^\bullet$$

- *Elements.* For all $\overline{A} \in \mathsf{Ty}_{\mathcal{C}^\star}(\overline{\Gamma})$ and $M, N \in \mathsf{Ty}_{\mathcal{C}^\star}(\overline{\Gamma})\overline{A}$ and $r \in \mathsf{Dim}_{\mathcal{C}^\star}(\overline{\Gamma})$, we must verify the following implication:

$$(\forall \gamma\gamma^\bullet.\ \overrightarrow{M^\bullet(\gamma.\gamma^* r = \varepsilon)\gamma^\bullet = N^\bullet(\gamma.\gamma^* r = \varepsilon)\gamma^\bullet}) \implies \forall \gamma\gamma^\bullet.\ M^\bullet\gamma\gamma^\bullet = N^\bullet\gamma\gamma^\bullet$$

This follows in an analogous way to the above from Lemma D.6, using the elementwise separation of $A^\bullet$. ◀

▶ **Lemma D.18.** *$\mathcal{C}^\star$ has type-case in the sense of Definition B.18.*

Proof. We fix the following glued data:

$$\overline{C} \in \mathsf{Ty}_{\mathcal{C}^\star}(\overline{\Gamma}) \qquad \overline{X} \in \mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma} \vdash \overline{\mathbf{U}_k}) \qquad \overline{M_\Pi}, \overline{M_\Sigma} \in \mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma}.\overline{\mathbf{U}_k}.\mathbf{\Pi}(\mathbf{q}, \overline{\mathbf{U}_k}) \vdash (\mathbf{p} \circ \mathbf{p})^* \overline{C})$$

$$\overline{M_{\mathbf{Eq}}} \in \mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma}.\overline{\mathbf{U}_k}.\overline{\mathbf{U}_k}.\mathbf{Eq}(\_.\overline{\mathbf{U}_k}, \mathbf{p}^*\mathbf{q}, \mathbf{q}).(\mathbf{p} \circ \mathbf{p})^*\mathbf{q}.(\mathbf{p} \circ \mathbf{p})^*\mathbf{q} \vdash (\mathbf{p} \circ \mathbf{p} \circ \mathbf{p} \circ \mathbf{p} \circ \mathbf{p})^*\overline{C})$$

$$\overline{M_{\mathbf{bool}}} \in \mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma} \vdash \overline{C}) \qquad \overline{M_{\mathbf{U}}} \in \mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma} \vdash \overline{C})$$

We need to exhibit an element $\mathbf{case}_{\mathbf{U}_k}(\overline{C}; \overline{X}; \overline{M_\Pi}; \overline{M_\Sigma}; \overline{M_{\mathbf{Eq}}}; \overline{M_{\mathbf{bool}}}; \overline{M_{\mathbf{U}}}) \in \mathsf{El}_{\mathcal{C}^\star}(\overline{\Gamma} \vdash \overline{C})$ with the specified computation and naturality rules. Inheriting the syntactic part from $\mathcal{C}$ as $\mathbf{case}_{\mathbf{U}_k}(C; X; M_\Pi; M_\Sigma; M_{\mathbf{Eq}}; M_{\mathbf{bool}}; M_{\mathbf{U}}) \in \mathsf{El}_{\mathcal{C}}(\Gamma \vdash C)$, it remains to define its realizer:

$$\big(\mathbf{case}_{\mathbf{U}_k}(\overline{C}; \overline{X}; \overline{M_\Pi}; \overline{M_\Sigma}; \overline{M_{\mathbf{Eq}}}; \overline{M_{\mathbf{bool}}}; \overline{M_{\mathbf{U}}})\big)^\bullet \gamma\gamma^\bullet =$$

$$\begin{cases} M_\Pi^\bullet\langle\gamma, A, \mathbf{lam}(B)\rangle((\gamma^\bullet, \mathtt{A}), \mathtt{B}) & \text{if } X^\bullet\gamma\gamma^\bullet = \mathtt{pi}(\mathtt{A}; \mathtt{B}) : \mathfrak{U}_k^\bullet \mathbf{\Pi}(A, B) \\ M_\Sigma^\bullet\langle\gamma, A, \mathbf{lam}(B)\rangle((\gamma^\bullet, \mathtt{A}), \mathtt{B}) & \text{if } X^\bullet\gamma\gamma^\bullet = \mathtt{sg}(\mathtt{A}; \mathtt{B}) : \mathfrak{U}_k^\bullet \mathbf{\Sigma}(A, B) \\ M_{\mathbf{Eq}}^\bullet & \text{if } X^\bullet\gamma\gamma^\bullet = \mathtt{eq}(\mathtt{A}; N_0^\bullet, N_1^\bullet) : \mathfrak{U}_k^\bullet \mathtt{eq}(i.A; N_0, N_1) \\ \quad \langle\gamma, (0/i)^\ddagger A, (1/i)^\ddagger A, \mathbf{plam}(i.A), N_0, N_1\rangle & \\ \quad (\gamma^\bullet, \mathtt{A0}, \mathtt{A1}, \mathtt{A}, N_0^\bullet, N_1^\bullet) & \\ M_{\mathbf{bool}}^\bullet\gamma\gamma^\bullet & \text{if } X^\bullet\gamma\gamma^\bullet = \mathtt{bool} \\ M_{\mathbf{U}}^\bullet\gamma\gamma^\bullet & \text{if } X^\bullet\gamma\gamma^\bullet = \mathtt{univ}_{k'} \end{cases}$$

The required equations follow by calculation. ◀

▶ **Corollary D.19.** *$\mathcal{C}^\star$ is a model of XTT and moreover, $\mathcal{C}^\star \xrightarrow{\pi_{\mathsf{syn}}} \mathcal{C}$ is a homomorphism of XTT-algebras.*