A parametricity-based formalization of semi-simplicial and semi-cubical sets

Hugo Herbelin
and Ramkumar Ramachandra

Université Paris Cité, Inria, CNRS, IRIF, Paris
Hugo.Herbelin@inria.fr and r@artagnon.com

***Abstract*—Constructions such as semi-simplicial and semi-cubical sets can be defined in the "usual way" as presheaves over respectively, the semi-simplex or semi-cube category, which we call *fibered* definitions, but also defined like in e.g. Voevodsky [1] or in previous work [2], as a dependently-typed construction, which we call *indexed*.**

This paper describes a uniform indexed characterization of both augmented semi-simplicial and semi-cubical sets arising respectively as unary and binary iterated parametricity-based constructions.

Additionally, our construction is fully formalized in Coq's dependent type theory.

I. Introduction

In the context of functional programming, Reynolds' parametricity [3] interprets types as relations characterizing the observational behavior of programs of this type. Parametricity can be iterated, and it has been noted that iterated Reynolds' parametricity has the structure of a cubical set [4], [5], [6], [7], [8]. We obtain a *unary* variant of Reynolds' *binary* parametricity by using predicates or families instead of relations or graphs, in which case, we obtain a form of realizability [9], [10], [6]. It has then been noted that iterated unary parametricity has the structure of an augmented simplicial set [a].

We exploit the connection between iterated unary parametricity and augmented simplicial sets, and between iterated binary parametricity and cubical sets to present a uniform construction of both augmented semi-simplicial and semi-cubical sets, generated by iterating the parametricity translation for type judgements. In contrast to the standard presheaf presentation of augmented semi-simplicial and semi-cubical sets [11], [12], [13], our uniform construction, which we call *ν-sets*, is *indexed*. That is, instead of having the set of augmented semi-simplices or semi-cubes in dimension $n+1$ *fibered* over the set of augmented semi-simplices or semi-cubes in dimension n, we consider families of augmented semi-simplices and semi-cubes indexed over their faces.

As a result, our work can be seen as contributing the following: it characterizes, will full details, how to technically define parametricity iterated in arbitrary dimensions; it provides a new example of indexed constructions of semi-simplicial and semi-cubical sets [1], [2], [14], useful in the direction of better understanding the technicality of coherence issues in defining "semi-simplicial types" [14], [15], [16], [17], as well as the dialectic between reasoning in Extensional Type Theory and Intensional Type Theory (as will be discussed in Section IV-B); it makes a step in the direction of the program initiated in [5] to develop parametricity-based models of parametric type theory [18], [19], [20] and cubical type theory [21], [22], [23] which are closer to the syntax of type theory and thus more liable to reflect definitional properties of type theory than presheaf-based cubical sets would do (compare e.g. to the loss of definitional properties when interpreting "indexed" dependent types of type theory as "fibrations" in models such as locally cartesian closed categories [24]).

The outline of the paper is as follows. We recall the definition of the augmented semi-simplicial and semi-cubical categories using a combinatorial presentation, that directly leads to our generalization to ν-sets, in II. We then proceed to explain the connection between the indexed representation and parametricity in III. Before spelling out the details of our formalization in type-theoretic language in IV-B and IV-C, we give intuitions in IV-A. We end with some of the finer details of our mechanization in IV-D.

See github.com/artagnon/bonak for our mechanization. The construction was conceived in Summer 2019, and the mechanization began in late 2019. A sketch of the construction was presented at the 2020 HoTT-UF workshop and the completion of the mechanization was reported at the TYPES 2022 conference.

II. Semi-simplicial and semi-cubical sets

In this section, we give a uniform definition of semi-simplicial and semi-cubical sets which we believe is folklore. The uniformity allows the definition of a notion of ν-sets that subsumes semi-simplicial and semi-cubical sets.

[a]Private communication with Hugo Moeneclaey and Thorsten Altenkirch

A. Augmented semi-simplicial sets

Augmented semi-simplicial sets are defined similarly to semi-simplicial sets, except that the connected components are additionally dependent on a "color". Conversely, semi-simplicial sets can be seen as augmented semi-simplicial sets over a singleton set of colors.

Let us associate dimension 0 to colors; then, points are dimension 1, lines are dimension 2, and so on. There is hence a shift by one when compared to semi-simplicial sets. We can then draw augmented semi-simplicial sets like semi-simplicial sets, except for this shift by one.

While ordinary semi-simplicial sets are presheaves over the semi-simplex category, augmented semi-simplicial sets are presheaves over, what we will define as, $\mathbf{\Delta}_+$. There are different ways to define $\mathbf{\Delta}_+$, up to equivalence, and we use a definition that will later straightforwardly extend to semi-cubical sets.

II-A.1 Notation [Finite sequences] We denote finite sequences by $[i_1, \dots i_n]$ for i_j ranging over some domain. In particular, the empty sequence is written $[]$ and we define $i :: [j_1, \dots, j_n]$ to be $[i, j_1, \dots, j_n]$.

II-A.2 Definition [$\mathbf{\Delta}_+$] The definition of $\mathbf{\Delta}_+$ is shown in Fig. 1. Note that, if $g \circ f$ is well-defined, then the length of f is less than that of g. It can be shown that composition is associative and that id is neutral.

II-A.3 Definition [$\mathsf{Set}_{\mathbf{\Delta}_+}$] We define the category of augmented semi-simplicial sets as the functor category:

$$\mathsf{Set}_{\mathbf{\Delta}_+} := \mathsf{Set}^{\mathbf{\Delta}_+^{\mathsf{op}}}$$

To provide examples, we define the standard augmented n-semi-simplex.

II-A.4 Definition [Δ_+^n] The standard augmented n-semi-simplex $\mathbf{\Delta}_+^n$ is defined as what is called the Yoneda embedding of $n \in \mathsf{Obj}(\mathbf{\Delta}_+)$:

$$\begin{aligned} \mathbf{\Delta}_+^n &: \mathsf{Set}_{\mathbf{\Delta}_+} \\ \mathbf{\Delta}_+^n(p) &:= \mathsf{Hom}(p, n) \\ \mathbf{\Delta}_+^n(f) &:= \lambda g.\, g \circ f \end{aligned}$$

The standard augmented 0-semi-simplex is a singleton made of one color (in our case black). Standard augmented n-semi-simplices for $n \geq 1$ have a geometric interpretation, and we illustrate them for dimensions 1, 2, and 3.

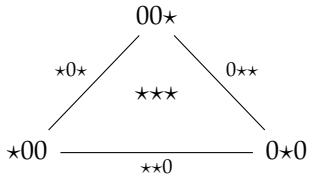
II-A.5 Example [Δ_+^1] The standard augmented 1-semi-simplex can be pictured as a point, colored black, corresponding to the unique morphism in $\mathsf{Hom}(0,1)$. This point is the identity in $\mathsf{Hom}(1,1)$; it is hence shown as a singleton $\star$.

$\star$

II-A.6 Example [Δ_+^2] The standard augmented 2-semi-simplex is drawn as two points, given by $\mathsf{Hom}(1,2)$, along with a line connecting them, given by $\mathsf{Hom}(2,2)$. We use the color black to denote the unique morphisms in $\mathsf{Hom}(0,1)$ and $\mathsf{Hom}(0,2)$.

$$\star 0 \xrightarrow{\star\star} 0\star$$

II-A.7 Example [Δ_+^3] $\mathbf{\Delta}_+^3$ is drawn as three points, given by $\mathsf{Hom}(1,3)$, three lines connecting them, given by $\mathsf{Hom}(2,3)$, and a triangular filler given by $\mathsf{Hom}(3,3)$.



More generally, the standard augmented $(n+1)$-semi-simplex can be obtained by taking a copy of the standard augmented n-semi-simplex serving as a base and gluing on top of it another copy lifted by one dimension. In the second copy, the color becomes an extra point, the points become lines connecting the points of the base to the extra point and so on. In particular, the components of the base are those of the standard augmented n-semi-simplex postfixed by 0 while the components of the lifted copy are postfixed by $\star$. Note that the components may be oriented by letting each n-dimensional component point to the $(n-1)$-dimensional component obtained by replacing the leftmost $\star$ of the n-dimensional component with 0.

B. Semi-cubical sets

Semi-cubical sets are defined like augmented semi-simplicial sets except that $\mathbf{\Delta}_+$ is replaced by $\square$ in which we take sequences of L, R and $\star$, instead of sequences of 0 and $\star$. They represent ordinary semi-cubical sets, with faces (and no augmentation).

II-B.1 Definition [$\square$] The definition of $\square$ is shown in Fig. 2. The symbols L and R indicate opposite faces of a cube.

Again, if $g \circ f$ is well-defined, then the length of f is less than that of g. It can be shown that composition is associative and that id is neutral.

II-B.2 Definition [$\mathsf{Set}_\square$] We define the category of semi-cubical sets as the functor category:

$$\mathsf{Set}_\square := \mathsf{Set}^{\square^{\mathsf{op}}}$$

II-B.3 Definition [$\square^n$] The standard semi-cube $\square^n$ is defined as the Yoneda embedding of $n \in \mathsf{Obj}(\square)$:

$$\begin{aligned} \square^n &: \mathsf{Set}_\square \\ \square^n(p) &:= \mathsf{Hom}(p, n) \\ \square^n(f) &:= \lambda g.\, g \circ f \end{aligned}$$

Standard n-semi-cubes have a geometric interpretation, which we illustrate for dimensions 0, 1, and 2.

$$
\begin{aligned}
\mathsf{Obj}(\mathbf{\Delta}_+) &:= \mathbb{N} \\
\mathsf{Hom}(p,n) &:= \{l \in [0,\star]^n \mid \text{number of } \star \text{ in } l = p\} \\
g \circ f &:= \begin{cases} f & \text{if } g = [] \\ 0 :: (g' \circ f) & \text{if } g = (0 :: g') \\ a :: (g' \circ f') & \text{if } g = (\star :: g'),\ f = (a :: f'), \text{ where } a = 0 \text{ or } \star \end{cases} \\
\mathsf{id} &:= [\star, \ldots, \star]\ n \text{ times for } \mathsf{id} \in \mathsf{Hom}(n,n)
\end{aligned}
$$

Fig. 1: Definition of $\mathbf{\Delta}_+$

$$
\begin{aligned}
\mathsf{Obj}(\square) &:= \mathbb{N} \\
\mathsf{Hom}(p,n) &:= \{l \in [L,R,\star]^n \mid \text{number of } \star \text{ in } l = p\} \\
g \circ f &:= \begin{cases} f & \text{if } g = [] \\ a :: (g' \circ f) & \text{if } g = (a :: g'), \text{where } a = L \text{ or } R \\ a :: (g' \circ f') & \text{if } g = (\star :: g'),\ f = (a :: f'), \text{ where } a = L,\ R, \text{ or } \star \end{cases} \\
\mathsf{id} &:= [\star, \ldots, \star]\ n \text{ times}
\end{aligned}
$$

Fig. 2: Definition of $\square$

II-B.4 Example [$\square^0$] $\square^0$ is $\mathsf{Hom}(0,0)$, or the singleton set of the empty sequence:

$$\{\}$$

II-B.5 Example [$\square^1$] $\square^1$ consists of two points, given by $\mathsf{Hom}(0,1)$ and a line, given by $\mathsf{Hom}(1,1)$.

$$L \xrightarrow{\star} R$$

II-B.6 Example [$\square^2$] $\square^2$ consists of four points, given by $\mathsf{Hom}(0,2)$, four lines connecting the four points, given by $\mathsf{Hom}(1,2)$, and a filler, given by $\mathsf{Hom}(2,2)$:

$$
\begin{array}{ccc}
LR & \xrightarrow{\star R} & RR \\
{\scriptstyle L\star}\downarrow & \star\star & \downarrow{\scriptstyle R\star} \\
LL & \xrightarrow[\star L]{} & RL
\end{array}
$$

More generally, the standard $(n+1)$-semi-cube can be obtained by taking two copies of the standard n-semi-cube serving as bottom and top face and connecting them on their border by a cylinder obtained as a third copy stretched in the new dimension. The bottom and top faces are obtained from the standard n-semi-cube by postfixing with respectively L and R while the cylinder is obtained by postfixing with $\star$. Note that the components can this time be oriented by letting each n-dimensional component go from the $(n-1)$-dimensional component obtained by replacing the leftmost $\star$ with L to the one obtained by replacing the leftmost $\star$ with R.

C. Generalization to ν-sets

Let us call ν-sets, the generalization of augmented semi-simplicial sets and semi-cubical sets. To obtain this, we extend $\mathbf{\Delta}_+$ and $\square$ in a straightforward manner into a category which we call ν-semi-shape category. The morphisms of the ν-semi-shape category are sequences of elements of a set ν of arbitrary cardinal, extended with $\star$, so that the following holds.

Value of ν	1	2
Interpretation	Augmented semi-simplicial types	Semi-cubical types

A ν-set is thus a contravariant functor ϕ from the ν-semi-shape category to Set and we call n-ν-semi-shape an element of $\phi(n)$. As in the augmented semi-simplicial and semi-cubical cases, the standard $(n+1)$-ν-semi-shape is obtained by connecting together ν copies of the standard n-ν-semi-shape with an extra copy stretched in the new dimension. We clarify in the next sections how this process of construction is similar to the parametricity translation developed for functional programming [3] and more generally for type theory [25], [26], [27], [18].

III. Indexed representation and parametricity

In this section, we introduce some preliminaries on interpreting sets in type theory, followed by a small section on the indexed presentation, before relating it to parametricity in III-C.

A. Preliminaries

Martin-Löf's Type theory [28], [29] is a logical formalism based on the notion of a *type* rather than that of a *set*. It can be seen as a foundation of mathematics alternative to set theory and is the core of several tools for the formalization of mathematics such as Agda [30], Coq [31] or Lean [32]. In type theory, propositions are types and proofs are programs. A particularity of type theory is also that types and programs, hence propositions and proofs also, are considered modulo an equational theory called definitional equality.

Type theory is a flexible formalism supporting different models. Some models are based on topological spaces, where equality is interpreted as path, and substitutivity of equality as transport [33]. These models support the univalence principle stating that equality of types mimics equivalence of types, leading to the development of Homotopy Type Theory [34]. In type theory, types are organized in a hierarchy of universes written Type_m for m a natural number. Main types in type theory are the types of dependent pairs, written $\Sigma a : A.\, B(a)$, the types of dependent functions, written $\Pi a : A.\, B(a)$, for A a type and $B(a)$ a type dependent on the inhabitant a of A, and the type of propositional equality, written $t = u$. We assume our type theory to also include a distinguished singleton type, written unit, and with inhabitant $*$, the type of boolean values, called bool, and the type of natural numbers. We also write hd and tl the projections of dependent pairs, and refl for reflexivity. Logical propositions being types themselves, we use Π to represent universal quantification and Σ to represent existential quantification.

A notion of sets can be recovered in each universe as HSet_m, denoting the subtype of Type_m for which paths are degenerated, what can be expressed by the property of Uniqueness of Identity Proofs (UIP). Technically, this is expressed as a structure equipping a domain Dom with the property UIP:

$$\mathsf{Dom} : \mathsf{Type}_m$$
$$\mathsf{UIP} : \Pi xy : \mathsf{Dom}.\, \Pi pq : x = y.\, p = q$$

In HSet_m, the following properties hold:

(i) UIP holds on the unit type, bool type, as well as all types of finite cardinal ν.
(ii) UIP propagates to Σ-types.
(iii) UIP propagates to Π-types, with some additional functional extensionality axioms.

Unless otherwise specified, we fix a universe level m and abbreviate Type_m as Type and HSet_m as HSet.

B. Fibered versus indexed representation

There are different ways to represent a family of sets, commonly known as the fibered and indexed representations. In type theory, they are equivalent, and this equivalence can be formulated as:

$$(\Sigma S : \mathsf{HSet}.(S \to T)) \simeq (T \to \mathsf{HSet})$$

Let us apply this correspondence to the case of semi-cubical sets. A semi-cubical set can be represented as a sequence of fibrations, together with appropriate coherence conditions.

$$X_0 : \mathsf{HSet} \;\begin{matrix}\leftarrow \delta^R - \\ \leftarrow \delta^L -\end{matrix}\; X_1 : \mathsf{HSet} \;\begin{matrix}\leftarrow \delta^{\star R} - \\ \leftarrow \delta^{\star L} - \\ \leftarrow \delta^{R\star} - \\ \leftarrow \delta^{L\star} -\end{matrix}\; X_2 : \mathsf{HSet} \qquad \dots$$

If we think of X_1 fibered over two copies of X_0 in an indexed way, we get X_1 of type $X_0 \to X_0 \to \mathsf{HSet}$. Continuing the process, we can think of X_2 fibered over four copies of X_1 themselves fibered over two copies of X_0 as a dependent type. We have a priori a dependency over eight copies of X_0 but these are actually four due to the coherence conditions on the functor (e.g. that $\delta^R \circ \delta^{R\star} = \delta^{(R\star)\circ(R)} = \delta^{RR} = \delta^{(\star R)\circ(R)} = \delta^R \circ \delta^{\star R}$, etc.). Based on these coherence conditions, the types of X_0, X_1, X_2 happen to have the form:

$$\begin{aligned} X_0 &: \mathsf{HSet} \\ X_1 &: X_0 \to X_0 \to \mathsf{HSet} \\ X_2 &: \Pi abcd.\quad X_1(a)(b) \to X_1(a)(c) \to X_1(b)(d) \\ &\qquad \to X_1(c)(d) \to \mathsf{HSet} \\ &\dots \end{aligned}$$

C. Relating to parametricity

The process of construction of the type of X_1 from that of X_0, and from the type of X_2 to that of X_1 in the last section, is similar to applying a binary parametricity translation and expecting the resulting translation to be inhabited. The parametricity which we consider interprets a type A by a graph $A_\star$ over this type, and a term $t : A$ as an edge in $A_\star(t)(t)$. In particular, HSet is interpreted as the graph $\mathsf{HSet}_\star$, which takes A_L and A_R in HSet and returns the type $A_L \to A_R \to \mathsf{HSet}$ of graphs over A_L and A_R. Also, for A interpreted by $A_\star$ and B interpreted by $B_\star$, a dependent function type $\Pi a : A.\, B$ is interpreted as the graph $(\Pi a : A.\, B)_\star$ that takes two functions f_L and f_R of type $\Pi a : A.\, B$, and expresses that these functions map related arguments in A to related arguments in B:

$$\begin{aligned} &(\Pi a : A.\, B)_\star(f_L)(f_R) \triangleq \\ &\Pi a_L : A.\, \Pi a_R : A.\, \Pi a_\star : A_\star(a_L)(a_R).\, B_\star(f_L(a_L))(f_R(a_R)) \end{aligned}$$

In particular, for $X : \mathsf{HSet}$, applying the parametricity translation is about canonically associating to X an inhabitant $X_\star$ of $\mathsf{HSet}_\star(X)(X)$ i.e. of $X \to X \to \mathsf{HSet}$. In turn, applying the parametricity translation to $X_\star$:

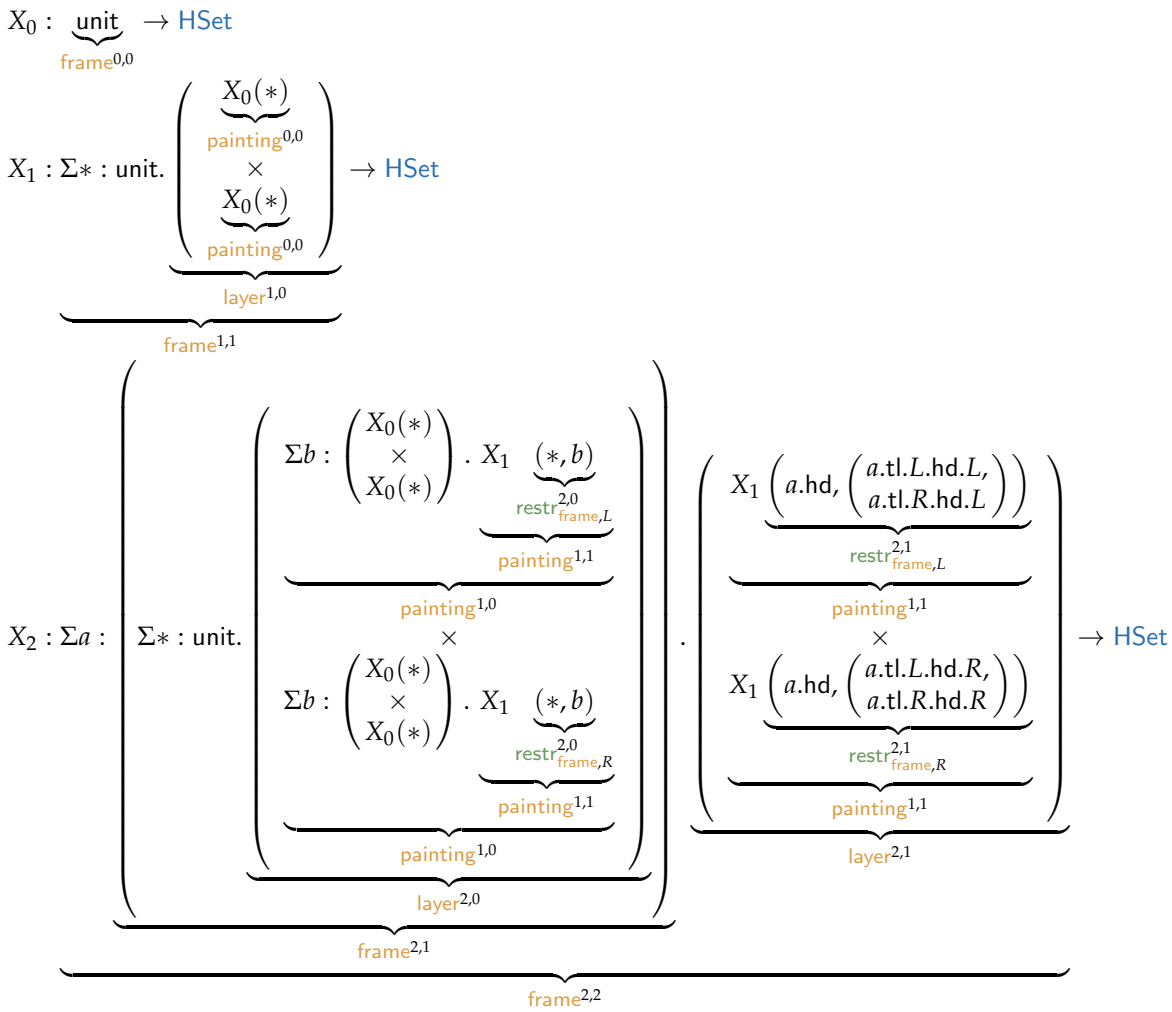


Fig. 3: Intuition for formal construction ($\nu = 2$)

$X \to X \to \mathsf{HSet}$ is about canonically associating to $X_\star$ an inhabitant $X_{\star\star}$ of $(X \to X \to \mathsf{HSet})_\star(X_\star)(X_\star)$ i.e. of:

$$\begin{aligned} &\Pi x_{LL} : X. \Pi x_{LR} : X. X_\star(x_{LL})(x_{LR}) \\ \to\ &\Pi x_{RL} : X. \Pi x_{RR} : X. X_\star(x_{RL})(x_{RR}) \\ \to\ &X_\star(x_{LL}, x_{RL}) \to X_\star(x_{LR})(x_{RR}) \to \mathsf{HSet} \end{aligned}$$

which hints us at how the sequence X_0, X_1, X_2 can be seen as a sequence of inhabitants of the iteration of binary parametricity applied to an initial $X : \mathsf{HSet}$:

$$\begin{array}{lcll} X_0 & \triangleq & X & : \mathsf{HSet} \\ X_1 & \triangleq & X_\star & : \mathsf{HSet}_\star(X)(X) \\ X_2 & \triangleq & X_{\star\star} & : (\mathsf{HSet}_\star(X)(X))_\star(X_\star)(X_\star) \\ \ldots & & & \end{array}$$

This tells us how the informal type given to X_2 in the previous section could be rephrased so that it comes as the instance of a general recipe characterizing the type of all X_i.

Notice, however, that the recipe obtained so far, $X_{n+1} : (S_n)_\star(X_n)(X_n)$ for $X_n : S_n$, applies parametricity on the *syntax* of the type of X_n. It does not directly yield a characterization of S_n as a function from n. Reformulating the recipe as an explicit recursive construction, without requiring an interpretation of the syntax of types, is the main outcome of this work, together with the mechanization and the uniform treatment of augmented semi-simplicial and semi-cubical sets by means of the generalization to ν-sets.

IV. Our construction

Our construction is highly dependent, with dependencies in inequality and equality proofs. It involves several involved details, some of them being visible only in the Coq formalization.

In Section IV-B and Tables I, II, III, IV, V, we give an informal, mathematical definition in Extensional Type Theory, which roughly corresponds to a level of reasoning "à

la set theory" where equalities are thought as "holding on the nose" without having to mention them explicitly once they are justified. In Section IV-C and Tables VI, VII, VIII, IX, X, we give the same mathematical definition in Intensional Type Theory, where transporting along coherence conditions is now made explicit. Adding this level of details helps to understand the specificities of coherence conditions.

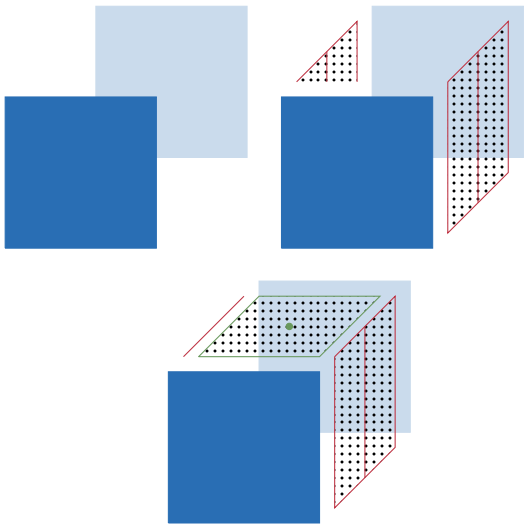
But first, we start with a section gradually and informally explaining the intuition of the construction. It can be read in parallel with the tables if wished.

A. Intuition for our formal construction

To assign types to $X_0, X_1, X_2, \dots$ in the above indexed representation, we give a recursive definition relying on building blocks, which we call frame, layer, and painting. A frame is a boundary of a standard form (simplex, cube, etc.), which we decompose into layer, and a painting corresponds to a filled frame. Some frame are full and we call them fullframe.

We assign every X_n the type $\mathsf{fullframe}^n \to \mathsf{HSet}$ uniformly, applying to the description above the isomorphism between $A \to B \to C$ and $A \times B \to C$, or between $\Pi a : A.\,(Ba \to C)$ and $(\Sigma a : A.\,Ba) \to C$. In particular, $\mathsf{fullframe}^n$ is a "telescope", i.e. a nesting of Σ-type. Let us now illustrate how we build $\mathsf{fullframe}^n$. To begin, let us set $\mathsf{fullframe}^0 = \mathsf{unit}$, so that the type HSet of X_0 can be equivalently formulated as $\mathsf{unit} \to \mathsf{HSet}$. Then, more generally, we let each $\mathsf{fullframe}^n$ consist of n layers, written $\mathsf{layer}^{n,p}$ with $p < n$, that we stack in order, starting from the unit type and writing $\mathsf{frame}^{n,p}$ for the p first layers of a $\mathsf{fullframe}^n$, so that $\mathsf{fullframe}^n$ is $\mathsf{frame}^{n,n}$. For instance, X_1 is made of one layer, so that it can be written as a Σ-type of an inhabitant of the unit and $\mathsf{layer}^{1,0}$, as shown in Fig. 3. Fig. 3 also mentions how the type of X_2 is structured.

Let us now illustrate the construction of $\mathsf{fullframe}^3$, necessary to build the type of X_3.



The figure on the left is $\mathsf{frame}^{3,1}$, on the right is $\mathsf{frame}^{3,2}$, and on the bottom is $\mathsf{frame}^{3,3}$, which is full. Further, $\mathsf{frame}^{3,1}$ is made of one layer, $\mathsf{layer}^{3,0}$, shown in blue, $\mathsf{frame}^{3,2}$ is made of one additional layer, $\mathsf{layer}^{3,1}$, shown in red, $\mathsf{frame}^{3,3}$ is made of one more layer, $\mathsf{layer}^{3,2}$, shown in green.

We illustrated here the cubical case, that is $\nu = 2$, but, in general, a $\mathsf{layer}^{n,p}$ is a product of ν $\mathsf{painting}^{n-1,p}$. A $\mathsf{painting}^{n,0}$ is a n-dimensional object corresponding to a filled $\mathsf{fullframe}^n$. More generally, a $\mathsf{painting}^{n,p}$ is an n-dimensional object which has the form of a $\mathsf{painting}^{n-p,0}$, thus of $(n-p)$-dimensional form, but shifted and living in dimensions p to n. Such $\mathsf{painting}^{n,p}$ fills a space framed by a partial $\mathsf{frame}^{n,p}$ so that, together, they form a filled $\mathsf{fullframe}^n$. For instance, in the picture, each of the two $\mathsf{painting}^{2,0}$ of $\mathsf{layer}^{3,0}$ is a filled blue square, each of the two $\mathsf{painting}^{2,1}$ of $\mathsf{layer}^{3,1}$ is the line shown in red, stretched into a partial square filling the partial frames made of respectively the left and right border of the blue square, and each of the two $\mathsf{painting}^{2,2}$ of $\mathsf{layer}^{3,2}$ is a point shown in green, stretched into a partial square filling the full frames made respectively of the upper and lower borders of the blue and red squares. A $\mathsf{painting}^{n,p}$ complements a $\mathsf{frame}^{n,p}$ by adding layers needed to form a $\mathsf{fullframe}^n$ and by filling the resulting $\mathsf{fullframe}^n$ with an inhabitant of X_n. Layers are added from dimension n to dimension p that is opposite to the order from 0 to p the $\mathsf{frame}^{n,p}$ are built, as shown below.

$$\mathsf{frame}^{n,p} \triangleq \Sigma a_n : (\dots(\Sigma * : \mathsf{unit}.\,\mathsf{layer}^{n,0})\dots).\,\mathsf{layer}^{n,p-1}$$
$$\mathsf{painting}^{n,p} \triangleq \Sigma l_p : \mathsf{layer}^{n,p}.(\dots(\Sigma l_n : \mathsf{layer}^{n,n-1}.\,X_n)\dots)$$

So far, we have not paid attention to the fact that we have a dependent type, shown as a Σ. Let us be more precise about this requirement. First, $\mathsf{fullframe}^n$ depends on all X_i up to $n-1$. So, we need to package up X_i, for $i < n$, into a nesting of Σ-types, which we abbreviate as $\nu\mathsf{Set}^{<n}$. This allows us to give the type $\nu\mathsf{Set}^{<n} \to \mathsf{HSet}$ to $\mathsf{fullframe}^n$. Then, for $D : \nu\mathsf{Set}^{<n}$, representing an initial prefix of $X_0, X_1, \dots X_{n-1}$, the indexed set X_n has type $\mathsf{fullframe}^n(D) \to \mathsf{HSet}$. Thus, $\mathsf{frame}^{n,p}$, $\mathsf{layer}^{n,p}$ and $\mathsf{painting}^{n,p}$ also depend on D. We can then refine the previous equation by showing the dependencies on D. In particular, X_n is just $D.\mathsf{tl}$.

$$\mathsf{frame}^{n,p}(D) \triangleq$$
$$\quad \Sigma a_n : (\dots(\Sigma * : \mathsf{unit}.\,\mathsf{layer}^{n,0}(D))\dots).\,\mathsf{layer}^{n,p-1}(D)$$
$$\mathsf{painting}^{n,p}(D) \triangleq$$
$$\quad \Sigma l_p : \mathsf{layer}^{n,p}(D).\,(\dots(\Sigma l_n : \mathsf{layer}^{n,n-1}(D).\,D.\mathsf{tl})\dots)$$

An extra refinement arises from the fact that each addition of a layer to a frame has to be glued onto the border of the partial frame built so far. So, each $\mathsf{layer}^{n,p}$ has to depend on $\mathsf{frame}^{n,p}$. We also need a way to characterize the border of the ν $\mathsf{painting}^{n-1,p}$ that compose a $\mathsf{layer}^{n,p}$, and this is $\mathsf{restr}^{n,p}_{\mathsf{frame},\epsilon,p}$, for all $\epsilon < \nu$.

$$\mathsf{frame}^{n,p}(D) \triangleq \Sigma d : (\ldots(\Sigma * : \mathsf{unit}.\, \mathsf{layer}^{n,0}(D)(*))\ldots).\, \mathsf{layer}^{n,p}(D)(d)$$

$$\mathsf{painting}^{n,p}(D)(d) \triangleq \Sigma l_p : \mathsf{layer}^{n,p}(D)(d).\, (\ldots(\Sigma l_n : \mathsf{layer}^{n,n-1}(D)(d, l_p, \ldots, l_{n-1}).\, D.\mathsf{tl}(d, l_p, \ldots, l_n))\ldots)$$

where $(d, l_p, \ldots, l_q)$ abbreviates $((\ldots(d, l_p), \ldots), l_q)$

Fig. 4: Refinement of definition of frame and painting

For instance, on the picture, the left and right $\mathsf{painting}^{2,1}$, shown in red, are laid on respectively the left and right borders of the blue boxes, and hence needs to depend on $\mathsf{frame}^{3,1}$. The left and right borders of the two blue boxes are extracted as $\mathsf{restr}^{2,1}_{\mathsf{frame},L}(D)(d)$ and $\mathsf{restr}^{2,1}_{\mathsf{frame},R}(D)(d)$. We can then refine the previous equation by showing the dependencies on d, as shown in Fig. 4.

When $\nu = 2$, the formation of layers from paintings amounts to:

$$\mathsf{layer}^{n,p}(D)(d) \triangleq \mathsf{painting}^{n-1,p}(D.\mathsf{hd})(\mathsf{restr}^{n,p}_{\mathsf{frame},L,p}(d)) \times \mathsf{painting}^{n-1,p}(D.\mathsf{hd})(\mathsf{restr}^{n,p}_{\mathsf{frame},R,p}(d))$$

The operation $\mathsf{restr}^{n,p}_{\mathsf{frame},\epsilon,q}$ indicates restriction on q layers of a frame, and the induction is on p, from 0 to q. In particular, $\mathsf{restr}^{n,p}_{\mathsf{frame},\epsilon,p}$ is a "full restriction". We define $\mathsf{restr}^{n,p}_{\mathsf{frame},\epsilon,p}(d)$ by recursion on the structure of a frame d, which necessitates definitions of $\mathsf{restr}^{n,p}_{\mathsf{layer},\epsilon,q}(d)(l)$ and $\mathsf{restr}^{n,p}_{\mathsf{painting},\epsilon,q}(d)(c)$, for l a layer and c a painting. The key case is $\mathsf{restr}^{n,p}_{\mathsf{painting},\epsilon,p}(d)(c)$, where c, a $\mathsf{painting}^{n,p}$, has necessarily the form of $((c_L, c_R), \_)$: $\mathsf{restr}^{n,p}_{\mathsf{painting},L,p}$ picks out c_L, a $\mathsf{painting}^{n-1,p}$, $\mathsf{restr}^{n,p}_{\mathsf{painting},R,p}$ picks out the c_R, also a $\mathsf{painting}^{n-1,p}$, and $\_$, a $\mathsf{painting}^{n,p+1}$, is discarded. There is one more difficulty, which we illustrate by writing down expected and actual types.

Given c_ω of type

$$c_\omega : \mathsf{painting}^{n-1,p}(D.\mathsf{hd})(\mathsf{restr}^{n-1,p}_{\mathsf{frame},\omega,q}(d))$$

$\mathsf{restr}^{n,p}_{\mathsf{layer},\epsilon}(d)(c_L, c_R)$ produces a layer, in which the ω-component has the type

$$\mathsf{painting}^{n-2,p}(D.\mathsf{hd}.\mathsf{hd})(\mathsf{restr}^{n-1,p}_{\mathsf{frame},\epsilon,q}(\mathsf{restr}^{n,p}_{\mathsf{frame},\omega,p}(d)))$$

while we expect a term of type

$$\mathsf{painting}^{n-2,p}(D.\mathsf{hd}.\mathsf{hd})(\mathsf{restr}^{n-1,p}_{\mathsf{frame},\omega,p}(\mathsf{restr}^{n,p}_{\mathsf{frame},\epsilon,q+1}(d)))$$

Hence, we need a coherence condition to commute the restrictions. Coherence conditions similar to this necessitate, what are shown as, $\mathsf{coh}_{\mathsf{frame}}$, $\mathsf{coh}_{\mathsf{layer}}$ and $\mathsf{coh}_{\mathsf{painting}}$ in tables in the next section. These are by induction on the structure of frame, layer and painting. Note that, for the construction in IV-C, we further need a 2-dimensional coherence condition, $\mathsf{coh2}_{\mathsf{frame}}$, for $\mathsf{coh}_{\mathsf{layer}}$.

B. Formal construction in ETT

We now present the construction in *extensional* type theory; i.e. in a type theory with the following reflection rule, where $=$ is propositional equality in some type and $\equiv$ is definitional equality [29]:

$$\frac{\Gamma \vdash p : t = u}{\Gamma \vdash t \equiv u}$$

For the presentation corresponding to the formalization in *intensional* type theory, see IV-C.

The definition is dispatched over tables I, II, III, IV and V. Table I describes a ν-set in indexed form, as a stream, coinductively representing the limit of n-truncated ν-sets. The n-truncated ν-sets are themselves described in II. In such stream, the nth component is a type dependent over a fullframe. The type fullframe is recursively defined in III, using the auxiliary definitions of layer and painting. The type layer and painting are dependent over $\mathsf{restr}_{\mathsf{frame}}$, and these restrictions are defined on IV. These restrictions are defined using auxiliary definitions of $\mathsf{restr}_{\mathsf{layer}}$ and $\mathsf{restr}_{\mathsf{painting}}$.

Notably, the definition of $\mathsf{restr}_{\mathsf{layer}}$ relies on a definitional equality expressing the commutation of the composition of $\mathsf{restr}_{\mathsf{frame}}$. This commutation is not provable by computation so we have to prove it propositionally before using the reflection rule. Proving this itself requires an induction on the dimension, and on the structure of frame, layer, and painting. This is what $\mathsf{coh}_{\mathsf{frame}}$ proves, as shown in the table V, using auxiliary definitions $\mathsf{coh}_{\mathsf{layer}}$ and $\mathsf{coh}_{\mathsf{painting}}$. Even though it looks independent of the other tables, $\mathsf{coh}_{\mathsf{frame}}$ has to be proved mutually with the definitions of frame, layer, painting, and their corresponding restrictions. More precisely, for a fixed n, the block of frame, $\mathsf{restr}_{\mathsf{frame}}$, and $\mathsf{coh}_{\mathsf{frame}}$ has to be mutually defined by induction on p. Also, each of painting, $\mathsf{restr}_{\mathsf{painting}}$, and $\mathsf{coh}_{\mathsf{painting}}$ is built by induction from p to n. The painting block at n relies on the frame block at n, but, the converse dependency is only on lower n, so this is well-founded. Note that layer, $\mathsf{restr}_{\mathsf{layer}}$ and $\mathsf{coh}_{\mathsf{layer}}$ are just abbreviations. We leave however implicit the exact way this mutual recursion can be formalized at this stage of the paper.

Most components of the construction takes inequality constraints as parameters, and we have left implicit that they are satisfied in the tables. Comparison of inequality proofs depends on the definition of inequality, which

we leave implicit, only assuming that we can pick a definition for which proofs of $m \leq n$ are provably unique and thus definitionally equal by the reflection rule.

The construction takes benefit of various provable equalities over proofs of equality being definitional by the reflection rule. This includes in particular the groupoid properties of equality. Notably, uniqueness of identity proofs holds in extensional type theory, so that any type is automatically an HSet. Also, we left implicit in table V the use of the isomorphism between $u = v$ and $\Sigma(p : u.\mathsf{hd} = v.\mathsf{hd}).(u.\mathsf{tl} = v.\mathsf{tl})$ for u and v in a Σ-type. In the same table, we also left implicit the use of the isomorphism between $f = g$ and $\Pi a : A.\, f(a) = g(a)$ for f and g in $\Pi a : A.\, B$, where it should be recalled that the right-to-left map, that is function extensionality, holds by default in extensional type theory.

Note that for a fixed constant n, the coherence conditions evaluate to a reflexivity proof, so that the construction evaluates to an effective sequence of types of iterated relations not mentioning them anymore. For instance, it produces Fig. 3 for $n = 2$ and $\nu = 2$.

$\nu\mathsf{Set}_m$		:	HSet_{m+1}
$\nu\mathsf{Set}_m$		$\triangleq$	$\nu\mathsf{Set}_m^{\geq 0}(*)$
$\nu\mathsf{Set}_m^{\geq n}$	$(D : \nu\mathsf{Set}_m^{<n})$	:	HSet_{m+1}
$\nu\mathsf{Set}_m^{\geq n}$	D	$\triangleq$	$\Sigma R : \nu\mathsf{Set}_m^{=n}(D).$ $\nu\mathsf{Set}_m^{\geq n+1}(D, R)$

TABLE I: **Main definition**

$\nu\mathsf{Set}_m^{<n}$		:	HSet_{m+1}
$\nu\mathsf{Set}_m^{<0}$		$\triangleq$	unit
$\nu\mathsf{Set}_m^{<n'+1}$		$\triangleq$	$\Sigma D : \nu\mathsf{Set}_m^{<n'}.\, \nu\mathsf{Set}_m^{=n'}(D)$
$\nu\mathsf{Set}_m^{=n}$	$(D : \nu\mathsf{Set}_m^{<n})$	:	HSet_m
$\nu\mathsf{Set}_m^{=n}$	D	$\triangleq$	$\mathsf{fullframe}_m^n(D) \to \mathsf{HSet}_m$

TABLE II: **Truncated ν-sets, the core**

C. Formal construction in ITT

We now present the definition in *intensional* type theory [29], [31], making explicit the need for transport along coherence proofs.

Tables VI, VII, and VIII are the same as tables I, II, and III, except that, in function applications, so as to be more informative, we make explicit all arguments, including those that can be inferred from the context.

The need for transport along a proof of commutation of $\mathsf{restr}_{\mathsf{frame}}$ in the definition of $\mathsf{restr}_{\mathsf{layer}}$ is made explicit as shown in table IX, where the arrow over $\mathsf{coh}_{\mathsf{frame}}$ indicates the direction of rewrite. The proof of $\mathsf{coh}_{\mathsf{frame}}$ is described on X, and it requires making explicit several rewritings which were invisible in extensional type theory. The commutation of $\mathsf{restr}_{\mathsf{layer}}$ lives in a type referring to $\mathsf{coh}_{\mathsf{frame}}$, so we need a transport along the commutation of $\mathsf{restr}_{\mathsf{frame}}$ in the statement of $\mathsf{coh}_{\mathsf{layer}}$. The proof of $\mathsf{coh}_{\mathsf{layer}}$ is the most involved proof in the construction. This is where the higher-dimensional coherence condition is needed. The exact formulation of this coherence condition is as follows.

$$\begin{aligned}
&\mathsf{coh}^{n,p}_{\mathsf{frame},\omega,\theta,r,p}(\mathsf{restr}^{n,p}_{\mathsf{frame},\epsilon,q+2}(d)) \bullet \\
&\mathsf{ap}\ \mathsf{restr}^{n,p}_{\mathsf{frame},\omega,r}\ (\mathsf{coh}^{n,p}_{\mathsf{frame},\epsilon,\theta,q+1,p}(d)) \bullet \\
&\mathsf{coh}^{n,p}_{\mathsf{frame},\epsilon,\omega,q,r}(\mathsf{restr}^{n,p}_{\mathsf{frame},\theta,p}(d)) = \\
&\mathsf{ap}\ \mathsf{restr}^{n,p}_{\mathsf{frame},\theta,p}\ (\mathsf{coh}^{n,p}_{\mathsf{frame},\epsilon,\omega,q+1,r+1}(d)) \bullet \\
&\mathsf{coh}^{n,p}_{\mathsf{frame},\epsilon,\theta,q,p}(\mathsf{restr}^{n,p}_{\mathsf{frame},\omega,r+1}(d)) \bullet \\
&\mathsf{ap}\ \mathsf{restr}^{n,p}_{\mathsf{frame},\epsilon,q}\ (\mathsf{coh}^{n,p}_{\mathsf{frame},\omega,\theta,r,p}(d))
\end{aligned}$$

where ap applies a function on two sides of an equality, and $\bullet$ is transitivity of equality. This property of equality proofs holds in HSet, and since our formalization is done in HSet, the term is trivially discharged.

Notice that each $\mathsf{restr}_{\mathsf{layer}}$ in the type of $\mathsf{coh}_{\mathsf{layer}}$ is hiding a $\mathsf{coh}_{\mathsf{frame}}$ rewrite: this makes a sum total of three $\mathsf{coh}_{\mathsf{frame}}$ rewrites on the left-hand side, and two $\mathsf{coh}_{\mathsf{frame}}$ rewrites on the right-hand side. In the proof term of $\mathsf{coh}_{\mathsf{layer}}$, $\mathsf{coh}_{\mathsf{painting}}$ has one $\mathsf{coh}_{\mathsf{frame}}$ rewrite on its left-hand side and zero on the right-hand side. This combined with the two terms of the form ap $\mathsf{coh}_{\mathsf{frame}}$ matches our expectation of three $\mathsf{coh}_{\mathsf{frame}}$ on the left-hand side and two $\mathsf{coh}_{\mathsf{frame}}$ on the right-hand side. Then, $\mathsf{coh2}_{\mathsf{frame}}$ can roughly be seen as a commutation of these $\mathsf{coh}_{\mathsf{frame}}$ terms.

Finally, let us explain $\mathsf{coh}_{\mathsf{painting}}$. The base case $p = r$ is the key case of the commutation of $\mathsf{restr}_{\mathsf{frame}}$, when one of the $\mathsf{restr}_{\mathsf{painting}}$ collapses, and the remaining equation holds trivially. The case of $p < r$ follows the structure of $\mathsf{restr}_{\mathsf{painting}}$ by induction.

Like in V, we leave implicit in X, the use of the isomorphism between the type of equalities of dependent pairs in a Σ-type with the Σ-type of equalities of the components of the pair, as well as the isomorphism between equality of functions with point-wise equality. Contrary to the ETT case, functional extensionality does not hold, so we assume it. However, the requirement of functional extensionality disappears if ν is finite. As for basic groupoid properties of equalities, which no longer hold definitionally, we do not mention them for simplicity.

The way recursion is implemented is still left implicit at this stage. See IV-D for the details.

Depending on how inequality on natural numbers is defined, different equalities may hold or not definitionally, such as transitivity of inequality or the computational properties of induction over inequality proofs. Details are given in IV-D.

Remark: If we were not working in HSet, but in HGpd, we would need to prove one more higher-dimensional coherence, and if we were working in Type, we would

$\mathsf{fullframe}^{n}$	$(D : \nu\mathsf{Set}_m^{<n})$	:	HSet_m
$\mathsf{fullframe}^{n}$	D	$\triangleq$	$\mathsf{frame}^{n,n}(D)$
$\mathsf{frame}^{n,p,p\leq n}$	$(D : \nu\mathsf{Set}_m^{<n})$	:	HSet_m
$\mathsf{frame}^{n,0}$	D	$\triangleq$	unit
$\mathsf{frame}^{n,p'+1}$	D	$\triangleq$	$\Sigma d : \mathsf{frame}^{n,p'}(D).\, \mathsf{layer}^{n,p'}(d)$
$\mathsf{layer}^{n,p,p<n}$	$\{D : \nu\mathsf{Set}_m^{<n}\}$ $(d : \mathsf{frame}^{n,p}(D))$	:	HSet_m
$\mathsf{layer}^{n,p}$	$D\ d$	$\triangleq$	$\Pi\omega.\mathsf{painting}^{n-1,p}(D.2)(\mathsf{restr}^{n,p}_{\mathsf{frame},\omega,p}(d))$
$\mathsf{painting}^{n,p,p\leq n}$	$(D : \nu\mathsf{Set}_m^{<n})$ $(E : \nu\mathsf{Set}_m^{=n}(D))$ $(d : \mathsf{frame}^{n,p}(D))$	:	HSet_m
$\mathsf{painting}^{n,p,p=n}$	$D\ E\ d$	$\triangleq$	$E(d)$
$\mathsf{painting}^{n,p,p<n}$	$D\ E\ d$	$\triangleq$	$\Sigma l : \mathsf{layer}^{n,p}(d).\, \mathsf{painting}^{n,p+1}(E)(d,l)$

TABLE III: frame, layer, **and** painting

$\mathsf{restr}^{n,p,p\leq q\leq n-1}_{\mathsf{frame},\epsilon,q}$	$\{D : \nu\mathsf{Set}^{<n}\}$ $(d : \mathsf{frame}^{n,p}(D))$	:	$\mathsf{frame}^{n-1,p}(D.1)$
$\mathsf{restr}^{n,0}_{\mathsf{frame},\epsilon,q}$	$D\ *$	$\triangleq$	$*$
$\mathsf{restr}^{n,p'+1}_{\mathsf{frame},\epsilon,q}$	$D\ (d,l)$	$\triangleq$	$(\mathsf{restr}^{n,p'}_{\mathsf{frame},\epsilon,q}(d), \mathsf{restr}^{n,p'}_{\mathsf{layer},\epsilon,q-1}(l))$
$\mathsf{restr}^{n,p,p\leq q\leq n-2}_{\mathsf{layer},\epsilon,q}$	$\{D : \nu\mathsf{Set}^{<n}\}$ $\{d : \mathsf{frame}^{n,p}(D)\}$ $(l : \mathsf{layer}^{n,p}(d))$	:	$\mathsf{layer}^{n-1,p}(\mathsf{restr}^{n,p}_{\mathsf{frame},\epsilon,q+1}(d))$
$\mathsf{restr}^{n,p}_{\mathsf{layer},\epsilon,q}$	$D\ d\ l$	$\triangleq$	$\lambda\omega.(\mathsf{restr}^{n-1,p}_{\mathsf{painting},\epsilon,q}(D.2)(l_\omega))$
$\mathsf{restr}^{n,p,p\leq q\leq n-1}_{\mathsf{painting},\epsilon,q}$	$(D : \nu\mathsf{Set}^{<n})$ $(E : \nu\mathsf{Set}^{=n}(D))$ $(d : \mathsf{frame}^{n,p}(D))$ $(c : \mathsf{painting}^{n,p}(E)(d))$	:	$\mathsf{painting}^{n-1,p}(D.2)(\mathsf{restr}^{n,p}_{\mathsf{frame},\epsilon,q+1}(d))$
$\mathsf{restr}^{n,p,p=q}_{\mathsf{painting},\epsilon,q}$	$D\ E\ d\ (l,\_)$	$\triangleq$	l_ϵ
$\mathsf{restr}^{n,p,p<q}_{\mathsf{painting},\epsilon,q}$	$D\ E\ d\ (l,c)$	$\triangleq$	$(\mathsf{restr}^{n,p}_{\mathsf{layer},\epsilon,q}(l), \mathsf{restr}^{n,p+1}_{\mathsf{painting},\epsilon,q}(E)(c))$

TABLE IV: q**-th projection of** restr**, or faces**

$\mathsf{coh}^{n,p,p\leq r\leq q\leq n-2}_{\mathsf{frame},\epsilon,\omega,q,r}$	$\{D : \nu\mathsf{Set}^{<n}\}$ $(d : \mathsf{frame}(D))$	:	$\mathsf{restr}^{n-1,p}_{\mathsf{frame},\epsilon,q}(\mathsf{restr}^{n,p}_{\mathsf{frame},\omega,r}(d))$ $= \mathsf{restr}^{n-1,p}_{\mathsf{frame},\omega,r}(\mathsf{restr}^{n,p}_{\mathsf{frame},\epsilon,q+1}(d))$
$\mathsf{coh}^{n,0}_{\mathsf{frame},\epsilon,\omega,q,r}$	$D\ *$	$\triangleq$	$\mathsf{refl}(*)$
$\mathsf{coh}^{n,p'+1}_{\mathsf{frame},\epsilon,\omega,q,r}$	$D\ (d,l)$	$\triangleq$	$(\mathsf{coh}^{n,p'}_{\mathsf{frame},\epsilon,\omega,q,r}(d), \mathsf{coh}^{n,p'}_{\mathsf{layer},\epsilon,\omega,q,r}(l))$
$\mathsf{coh}^{n,p,p<r\leq q\leq n-2}_{\mathsf{layer},\epsilon,\omega,q,r}$	$(D : \nu\mathsf{Set}^{<n})$ $\{d : \mathsf{frame}(D)\}$ $(l : \mathsf{layer}(d))$	:	$\mathsf{restr}^{n-1,p}_{\mathsf{layer},\epsilon,q}(\mathsf{restr}^{n,p}_{\mathsf{layer},\omega,r}(l))$ $= \mathsf{restr}^{n-1,p}_{\mathsf{layer},\omega,r}(\mathsf{restr}^{n,p}_{\mathsf{layer},\epsilon,q+1}(l))$
$\mathsf{coh}^{n,p}_{\mathsf{layer},\epsilon,\omega,q,r}$	$D\ d\ l$	$\triangleq$	$\lambda\theta.\ \mathsf{coh}^{n-1,p}_{\mathsf{painting},\epsilon,\omega,q-1,r-1}(D.2)(l_\theta)$
$\mathsf{coh}^{n,p,p\leq r\leq q\leq n-2}_{\mathsf{painting},\epsilon,\omega,q,r}$	$(D : \nu\mathsf{Set}^{<n})$ $(E : \nu\mathsf{Set}^{=n}(D))$ $(d : \mathsf{frame}(D))$ $(c : \mathsf{painting}(E)(d))$	:	$\mathsf{restr}^{n-1,p}_{\mathsf{painting},\epsilon,q}(D.2)(\mathsf{restr}^{n,p}_{\mathsf{painting},\omega,r}(E)(c))$ $= \mathsf{restr}^{n-1,p}_{\mathsf{painting},\omega,r}(D.2)(\mathsf{restr}^{n,p}_{\mathsf{painting},\epsilon,q+1}(E)(c))$
$\mathsf{coh}^{n,p,p=r}_{\mathsf{painting},\epsilon,\omega,q,r}$	$D\ E\ d\ (l,\_)$	$\triangleq$	$\mathsf{refl}(\mathsf{restr}^{n-1,p}_{\mathsf{painting},\epsilon,q-1}(D.2)(l_\epsilon))$
$\mathsf{coh}^{n,p,p<r}_{\mathsf{painting},\epsilon,\omega,q,r}$	$D\ E\ d\ (l,c)$	$\triangleq$	$(\mathsf{coh}^{n,p}_{\mathsf{layer},\epsilon,\omega,q,r}(l), \mathsf{coh}^{n,p+1}_{\mathsf{painting},\epsilon,\omega,q,r}(E)(c))$

TABLE V: **Commutation of** q**-th projection and** r**-th projection, or coherence conditions**

need to prove arbitrarily many higher-dimensional coherences. Here, HGpd is the subset of types A such that for all x and y in A, $x = y$ is in HSet. See [2], [16], [35] for a discussion on the need for recursive higher-dimensional coherence conditions in formulating higher-dimensional structures in type theory.

$\nu\mathsf{Set}_m$		:	HSet_{m+1}
$\nu\mathsf{Set}_m$		$\triangleq$	$\nu\mathsf{Set}^{\geq 0}(*)$
$\nu\mathsf{Set}^{\geq n}$	$(D : \nu\mathsf{Set}^{<n})$	:	HSet_{m+1}
$\nu\mathsf{Set}^{\geq n}$	D	$\triangleq$	$\Sigma R : \nu\mathsf{Set}^{=n}(D).$ $\nu\mathsf{Set}^{\geq n+1}(D, R)$

TABLE VI: **Main definition**

$\nu\mathsf{Set}^{<n}$		:	HSet_{m+1}
$\nu\mathsf{Set}^{<0}$		$\triangleq$	unit
$\nu\mathsf{Set}^{<n'+1}$		$\triangleq$	$\Sigma D : \nu\mathsf{Set}^{<n'}. \nu\mathsf{Set}^{=n'}(D)$
$\nu\mathsf{Set}^{=n}$	$(D : \nu\mathsf{Set}^{<n})$	:	HSet_m
$\nu\mathsf{Set}^{=n}$	D	$\triangleq$	$\mathsf{fullframe}^n_m(D) \to \mathsf{HSet}_m$

TABLE VII: **Truncated ν-sets, the core**

D. Details on the mechanization

Since the construction shown in the previous sections is by induction on n, and dependencies are on lower n and $p < n$, one would imagine formalizing this using well-founded induction in dependent type theory. We initially tried this approach: we had terms dependent on the proofs of the case distinction of $n' \leq n$ implies $n' < n$ or $n' = n$, and these proofs did not have definitional computational rule; it started to be necessary to reason propositionally on the computational property of the case distinction, and it eventually turned out to be unmanageable. Hence, we chose a different route: in practice, since $\mathsf{restr}^n_{\mathsf{frame}}$ depends on frame^n and frame^{n-1}, while $\mathsf{coh}^n_{\mathsf{frame}}$ depends on frame^n, frame^{n-1}, and frame^{n-2}, we only need to keep track of three "levels", and we built separate data structures for the levels, with dependencies. More concretely, we build the ten definitions shown in the tables by induction, and this is part of the definition of a larger record. The other fields of the record are frame, layer, $\mathsf{painting}$ at levels $n-1$ and $n-2$, $\mathsf{restr}_{\mathsf{frame}}$, $\mathsf{restr}_{\mathsf{layer}}$, and $\mathsf{restr}_{\mathsf{painting}}$ at level $n-1$, and equations to recall the definitions of these objects at lower levels.

The entire construction relies on inequalities over natural numbers, and we use two different definitions of $\leq$ addressing different concerns in our formalization. In order to build our first variant, we present an intermediate "recursive definition", phrased as:

```
Fixpoint leR (n m : nat) : SProp :=
match n, m with
| O, _ => STrue
| S n, O => SFalse
| S n, S m => leR n m
end.
```

Here SProp is a definitionally proof-irrelevant impredicative universe at the bottom of the universe hierarchy [36]. By placing the definition in SProp, we have definitional equality of all inequality proofs. For the purpose of unification, however, this definition does not go far enough. Consider the unification problems:

```
leR_trans ?p leR_refl = ?p
leR_trans leR_refl ?p = ?p
```

where `leR_trans` is transitivity, `leR_refl` is reflexivity, and `?p` is an existential variable. These two problems definitionally hold in SProp, but equalizing them does not solve the existential. For unification to be useful in inferring existentials, we present our first variant of $\leq$, which we dub as the "Yoneda variant":

```
Definition leY n m :=
  forall p, leR p n -> leR p m.
```

This definition is an improvement over `leR` since reflexivity is now definitionally the neutral element of transitivity, and associativity of transitivity also holds definitionally. Although it significantly eases our proof, there are some instances where unification is unable to solve the existentials, and we have to provide them explicitly.

The second variant of $\leq$, the "inductive variant", is phrased as:

```
Inductive leI : nat -> nat -> Type :=
| leI_refl n : n <~ n
| leI_down {n p} : p.+1 <~ n -> p <~ n
where "n <~ m" := (leI n m) : nat_scope.
```

Compared to `leY`, `leI` has no proof-irrelevance properties. This definition is specially crafted for painting, where we have to reason inductively from $p \leq n$ to n. In our usage, we have lemmas `leY_of_leI` and `leI_of_leY` in order to equip `leY` with the induction scheme of `leI`. The resulting induction scheme has computational rules holding propositionally.

V. Future work

In the cubical case, we expect the construction to eventually provide a model of (some version of) parametric type theory [19], [20] by adding degeneracies, a hierarchy of universes (as sketched e.g. in a talk by Herbelin at the HoTT-UF workshop for the bridge case [37]), as well as reasoning modulo permutations [12].

By equipping the universe construction with a structure of equivalences, as suggested along the lines of Altenkirch and Kaposi [5], we expect the construction to be able to serve as a basis for syntactic models of various versions of cubical type theory [21], [22], [23], saving the detour via the fibered approach inherent to usual presheaf models. This would a priori preserve definitional properties which may be lost when detouring via presheaves. In particular, we conjecture being able to

$\mathsf{fullframe}_m^n$	$(D : \nu\mathsf{Set}^{<n})$	:	HSet_m
$\mathsf{fullframe}_m^n$	D	$\triangleq$	$\mathsf{frame}_m^{n,n}(D)$
$\mathsf{frame}_m^{n,p,p\leq n}$	$(D : \nu\mathsf{Set}^{<n})$	:	HSet_m
$\mathsf{frame}_m^{n,0}$	D	$\triangleq$	unit
$\mathsf{frame}_m^{n,p'+1}$	D	$\triangleq$	$\Sigma d : \mathsf{frame}_m^{n,p'}(D).\, \mathsf{layer}_m^{n,p'}(D)(d)$
$\mathsf{layer}_m^{n,p,p<n}$	$(D : \nu\mathsf{Set}^{<n})$ $(d : \mathsf{frame}_m^{n,p}(D))$	:	HSet_m
$\mathsf{layer}_m^{n,p}$	$D\ d$	$\triangleq$	$\Pi\omega.\mathsf{painting}_m^{n-1,p}(D.1)(D.2)(\mathsf{restr}^{n,p}_{\mathsf{frame},m,\omega,p}(D)(d))$
$\mathsf{painting}_m^{n,p,p\leq n}$	$(D : \nu\mathsf{Set}^{<n})$ $(E : \nu\mathsf{Set}^{=n}(D))$ $(d : \mathsf{frame}_m^{n,p}(D))$	:	HSet_m
$\mathsf{painting}_m^{n,p,p=n}$	$D\ E\ d$	$\triangleq$	$E(d)$
$\mathsf{painting}_m^{n,p,p<n}$	$D\ E\ d$	$\triangleq$	$\Sigma l : \mathsf{layer}_m^{n,p}(D)(d).\, \mathsf{painting}_m^{n,p+1}(D)(E)(d,l)$

TABLE VIII: frame, layer, **and** painting

justify univalence holding definitionally. Our approach would also definitively ground cubical type theory in iterated parametricity.

References

[1] V. Voevodsky, "Semi-simplicial types," Nov 2012, shortly described in Section 8 of [2].

[2] H. Herbelin, "A dependently-typed construction of semi-simplicial types," *Mathematical Structures in Computer Science*, vol. 25, no. 5, pp. 1116–1131, 2015.

[3] J. C. Reynolds, "Types, abstraction and parametric polymorphism," in *Information Processing 83, Proceedings of the IFIP 9th World Computer Congress, Paris, France, September 19-23, 1983*, R. E. A. Mason, Ed. North-Holland/IFIP, 1983, pp. 513–523.

[4] P. Johann and K. Sojakova, "Cubical categories for higher-dimensional parametricity," *CoRR*, vol. abs/1701.06244, 2017. [Online]. Available: http://arxiv.org/abs/1701.06244

[5] T. Altenkirch and A. Kaposi, "Towards a cubical type theory without an interval," in *21st International Conference on Types for Proofs and Programs, TYPES 2015, May 18-21, 2015, Tallinn, Estonia*, ser. LIPIcs, T. Uustalu, Ed., vol. 69. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015, pp. 3:1–3:27. [Online]. Available: https://doi.org/10.4230/LIPIcs.TYPES.2015.3

[6] G. Moulin, "Internalizing parametricity," Ph.D. dissertation, Department of Computer Science and Engineering, Chalmers University of Technology, 2016.

[7] H. Moeneclaey, "Parametricity and semi-cubical types," in *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*. IEEE, 2021, pp. 1–11. [Online]. Available: https://doi.org/10.1109/LICS52264.2021.9470728

[8] ——, "Cubical models are cofreely parametric," Ph.D. dissertation, Université Paris Cité, 2022.

[9] J.-P. Bernardy and G. Moulin, "A computational interpretation of parametricity," in *2012 27th Annual IEEE Symposium on Logic in Computer Science*. IEEE, 2012, pp. 135–144.

[10] M. Lasson, "Réalisabilité et paramétricité dans les systèmes de types purs," Ph.D. Thesis, Université de Lyon, Nov. 2014.

[11] G. Friedman, "An elementary illustrated introduction to simplicial sets," *arXiv preprint arXiv:0809.4221*, 2008. [Online]. Available: https://arxiv.org/abs/0809.4221v7

[12] M. Grandis and L. Mauri, "Cubical sets and their site," *Theory and Applications of Categories*, vol. 11, pp. 185–211, 2003.

[13] U. Buchholtz and E. Morehouse, "Varieties of cubical sets," in *International Conference on Relational and Algebraic Methods in Computer Science*. Springer, 2017, pp. 77–92.

[14] F. Part and Z. Luo, "Semi-simplicial types in logic-enriched homotopy type theory," *CoRR*, vol. abs/1506.04998, 2015. [Online]. Available: http://arxiv.org/abs/1506.04998

[15] M. Shulman, "Homotopy type theory should eat itself (but so far, it's too big to swallow)," Mar 2014, blog post, homotopytypetheory.org/2014/03/03/hott-should-eat-itself.

[16] T. Altenkirch, P. Capriotti, and N. Kraus, "Extending homotopy type theory with strict equality," in *25th EACSL Annual Conference on Computer Science Logic, CSL 2016, August 29 - September 1, 2016, Marseille, France*, ser. LIPIcs, J. Talbot and L. Regnier, Eds., vol. 62. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016, pp. 21:1–21:17. [Online]. Available: https://doi.org/10.4230/LIPIcs.CSL.2016.21

[17] N. Kraus and C. Sattler, "Space-valued diagrams, type-theoretically," *arXiv preprint arXiv:1704.04543*, 2017.

[18] J. Bernardy, T. Coquand, and G. Moulin, "A presheaf model of parametric type theory," *Electr. Notes Theor. Comput. Sci.*, vol. 319, pp. 67–82, 2015. [Online]. Available: https://doi.org/10.1016/j.entcs.2015.12.006

[19] A. Nuyts, A. Vezzosi, and D. Devriese, "Parametric quantifiers for dependent type theory," *Proc. ACM Program. Lang.*, vol. 1, no. ICFP, aug 2017. [Online]. Available: https://doi.org/10.1145/3110276

[20] E. Cavallo and R. Harper, "Internal Parametricity for Cubical Type Theory," in *28th EACSL Annual Conference on Computer Science Logic (CSL 2020)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), M. Fernández and A. Muscholl, Eds., vol. 152. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020, pp. 13:1–13:17. [Online]. Available: https://drops.dagstuhl.de/opus/volltexte/2020/11656

[21] M. Bezem, T. Coquand, and S. Huber, "A model of type theory in cubical sets," in *19th International Conference on Types for Proofs and Programs, TYPES 2013, April 22-26, 2013, Toulouse, France*, ser. LIPIcs, R. Matthes and A. Schubert, Eds., vol. 26. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013, pp. 107–128. [Online]. Available: https://doi.org/10.4230/LIPIcs.TYPES.2013.107

[22] C. Cohen, T. Coquand, S. Huber, and A. Mörtberg, "Cubical Type Theory: A Constructive Interpretation of the Univalence Axiom," in *TYPES 2015*, ser. LIPIcs, T. Uustalu, Ed., vol. 69. Schloss Dagstuhl, 2018, pp. 5:1–5:34. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2018/8475

[23] C. Angiuli, G. Brunerie, T. Coquand, R. Harper, K. H. (Favonia), and D. R. Licata, "Syntax and models of cartesian cubical type theory," *Math. Struct. Comput. Sci.*, vol. 31, no. 4, pp. 424–468, 2021. [Online]. Available: https://doi.org/10.1017/S0960129521000347

[24] P.-L. Curien, R. Garner, and M. Hofmann, "Revisiting the categorical interpretation of dependent type theory," *Theoretical Computer Science*, vol. 546, pp. 99–119, 2014, models of Interaction: Essays in Honour of Glynn Winskel. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304397514001789

[25] J. Bernardy, P. Jansson, and R. Paterson, "Parametricity and dependent types," in *Proceeding of the 15th ACM SIGPLAN*

$\mathsf{restr}^{n,p,p\le q\le n-1}_{\mathsf{frame},m,\epsilon,q}$	$(D:\nu\mathsf{Set}^{<n})$ $(d:\mathsf{frame}^{n,p}_m(D))$	:	$\mathsf{frame}^{n-1,p}_m(D.1)$
$\mathsf{restr}^{n,0}_{\mathsf{frame},m,\epsilon,q}$	$D\ *$	$\triangleq$	$*$
$\mathsf{restr}^{n,p'+1}_{\mathsf{frame},m,\epsilon,q}$	$D\ (d,l)$	$\triangleq$	$(\mathsf{restr}^{n,p'}_{\mathsf{frame},m,\epsilon,q}(D)(d),\mathsf{restr}^{n,p'}_{\mathsf{layer},m,\epsilon,q-1}(D)(d)(l))$
$\mathsf{restr}^{n,p,p\le q\le n-2}_{\mathsf{layer},m,\epsilon,q}$	$(D:\nu\mathsf{Set}^{<n})$ $(d:\mathsf{frame}^{n,p}_m(D))$ $(l:\mathsf{layer}^{n,p}_m(D)(d))$	:	$\mathsf{layer}^{n-1,p}_m(D.1)(\mathsf{restr}^{n,p}_{\mathsf{frame},m,\epsilon,q+1}(D)(d))$
$\mathsf{restr}^{n,p}_{\mathsf{layer},m,\epsilon,q}$	$D\ d\ l$	$\triangleq$	$\lambda\omega.(\overrightarrow{\mathsf{coh}^{n,p}_{\mathsf{frame},m,\epsilon,\omega,q,p}(D)(d)}(\mathsf{restr}^{n-1,p}_{\mathsf{painting},m,\epsilon,q}(D.1)(D.2)(\mathsf{restr}^{n,p}_{\mathsf{frame},m,\omega,p}(D)(d))(l_\omega)))$
$\mathsf{restr}^{n,p,p\le q\le n-1}_{\mathsf{painting},m,\epsilon,q}$	$(D:\nu\mathsf{Set}^{<n})$ $(E:\nu\mathsf{Set}^{=n}(D))$ $(d:\mathsf{frame}^{n,p}_m(D))$ $(c:\mathsf{painting}^{n,p}_m(D)(E)(d))$	:	$\mathsf{painting}^{n-1,p}_m(D.1)(D.2)(\mathsf{restr}^{n,p}_{\mathsf{frame},m,\epsilon,q+1}(D)(d))$
$\mathsf{restr}^{n,p,p=q}_{\mathsf{painting},m,\epsilon,q}$	$D\ E\ d\ (l,\_)$	$\triangleq$	l_ϵ
$\mathsf{restr}^{n,p,p<q}_{\mathsf{painting},m,\epsilon,q}$	$D\ E\ d\ (l,c)$	$\triangleq$	$(\mathsf{restr}^{n,p}_{\mathsf{layer},m,\epsilon,q}(D)(d)(l),\mathsf{restr}^{n,p+1}_{\mathsf{painting},m,\epsilon,q}(D)(E)(d,l)(c))$

TABLE IX: q**-th projection of** restr**, or faces**

$\mathsf{coh}^{n,p,p\le r\le q\le n-2}_{\mathsf{frame},m,\epsilon,\omega,q,r}$	$(D:\nu\mathsf{Set}^{<n})$ $(d:\mathsf{frame}^{n,p}_m(D))$	:	$\mathsf{restr}^{n-1,p}_{\mathsf{frame},m,\epsilon,q}(D.1)(\mathsf{restr}^{n,p}_{\mathsf{frame},m,\omega,r}(D)(d))$ $=\mathsf{restr}^{n-1,p}_{\mathsf{frame},m,\omega,r}(D.1)(\mathsf{restr}^{n,p}_{\mathsf{frame},m,\epsilon,q+1}(D)(d))$
$\mathsf{coh}^{n,0}_{\mathsf{frame},m,\epsilon,\omega,q,r}$	$D\ *$	$\triangleq$	$\mathsf{refl}(*)$
$\mathsf{coh}^{n,p'+1}_{\mathsf{frame},m,\epsilon,\omega,q,r}$	$D\ (d,l)$	$\triangleq$	$(\mathsf{coh}^{n,p'}_{\mathsf{frame},m,\epsilon,\omega,q,r}(D)(d),\mathsf{coh}^{n,p'}_{\mathsf{layer},m,\epsilon,\omega,q,r}(D)(d)(l))$
$\mathsf{coh}^{n,p,p<r\le q\le n-2}_{\mathsf{layer},m,\epsilon,\omega,q,r}$	$(D:\nu\mathsf{Set}^{<n})$ $(d:\mathsf{frame}^{n,p}_m(D))$ $(l:\mathsf{layer}^{n,p}_m(D)(d))$	:	$\overrightarrow{\mathsf{coh}^{n,p}_{\mathsf{frame},m,\epsilon,\omega,q,r}(D)(d)}(\mathsf{restr}^{n-1,p}_{\mathsf{layer},m,\epsilon,q}(D.1)(\mathsf{restr}^{n,p}_{\mathsf{frame},m,\omega,r}(D)(d))(\mathsf{restr}^{n,p}_{\mathsf{layer},m,\omega,r}(D)(d)(l)))$ $=\mathsf{restr}^{n-1,p}_{\mathsf{layer},m,\omega,r}(D.1)(\mathsf{restr}^{n,p}_{\mathsf{frame},m,\epsilon,q+1}(D)(d))(\mathsf{restr}^{n,p}_{\mathsf{layer},m,\epsilon,q+1}(D)(d)(l))$
$\mathsf{coh}^{n,p}_{\mathsf{layer},m,\epsilon,\omega,q,r}$	$D\ d\ l$	$\triangleq$	$\overrightarrow{\lambda\theta.(\overrightarrow{\mathsf{coh2}^{n,p}_{\mathsf{frame},m,\epsilon,\omega,\theta,q,r}(d)})}(\mathsf{ap}\ (\overrightarrow{\mathsf{coh}^{n-1,p}_{\mathsf{frame},m,\omega,\theta,r,p}(D.1)(\mathsf{restr}^{n,p}_{\mathsf{frame},m,\epsilon,q+2}(D)(d))}))$ $(\mathsf{ap}\ (\overrightarrow{\mathsf{restr}^{n-1,p}_{\mathsf{frame},m,\omega,r}(D)(\mathsf{coh}^{n,p}_{\mathsf{frame},m,\epsilon,\theta,q+1,p})}))\ \mathsf{coh}^{n-1,p}_{\mathsf{painting},m,\epsilon,\omega,q-1,r-1}(D.2)(l_\theta)$
$\mathsf{coh}^{n,p,p\le r\le q\le n-2}_{\mathsf{painting},m,\epsilon,\omega,q,r}$	$(D:\nu\mathsf{Set}^{<n})$ $(E:\nu\mathsf{Set}^{=n}(D))$ $(d:\mathsf{frame}^{n,p}_m(D))$ $(c:\mathsf{painting}^{n,p}_m(D)(E)(d))$	:	$\overrightarrow{\mathsf{coh}^{n,p}_{\mathsf{frame},m,\epsilon,\omega,q,r}(D)(d)}(\mathsf{restr}^{n-1,p}_{\mathsf{painting},m,\epsilon,q}(D.1)(D.2)(\mathsf{restr}^{n,p}_{\mathsf{frame},m,\omega,r}(D)(d))(\mathsf{restr}^{n,p}_{\mathsf{painting},m,\omega,r}(D)(E)(d)(c)))$ $=\mathsf{restr}^{n-1,p}_{\mathsf{painting},m,\omega,r}(D.1)(D.2)(\mathsf{restr}^{n,p}_{\mathsf{frame},m,\epsilon,q+1}(D)(d))(\mathsf{restr}^{n,p}_{\mathsf{painting},m,\epsilon,q+1}(D)(E)(d)(c))$
$\mathsf{coh}^{n,p,p=r}_{\mathsf{painting},m,\epsilon,\omega,q,r}$	$D\ E\ d\ (l,\_)$	$\triangleq$	$\mathsf{refl}(\mathsf{restr}^{n-1,p}_{\mathsf{painting},m,\epsilon,q-1}(D.1)(D.2)(\mathsf{restr}^{n,p}_{\mathsf{frame},m,\omega,p}(D)(d))(l_\epsilon))$
$\mathsf{coh}^{n,p,p<r}_{\mathsf{painting},m,\epsilon,\omega,q,r}$	$D\ E\ d\ (l,c)$	$\triangleq$	$(\mathsf{coh}^{n,p}_{\mathsf{layer},m,\epsilon,\omega,q,r}(D)(d)(l),\mathsf{coh}^{n,p+1}_{\mathsf{painting},m,\epsilon,\omega,q,r}(D)(E)(d,l)(c))$

TABLE X: **Commutation of** q**-th projection and** r**-th projection, or coherence conditions**

international conference on Functional programming, ICFP 2010, Baltimore, Maryland, USA, September 27-29, 2010, P. Hudak and S. Weirich, Eds. ACM, 2010, pp. 345–356. [Online]. Available: http://doi.acm.org/10.1145/1863543.1863592

[26] J. Bernardy and M. Lasson, "Realizability and parametricity in pure type systems," in *Foundations of Software Science and Computational Structures - 14th International Conference, FOSSACS 2011, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2011, Saarbrücken, Germany, March 26-April 3, 2011. Proceedings*, ser. Lecture Notes in Computer Science, vol. 6604. Springer, 2011, pp. 108–122. [Online]. Available: https://doi.org/10.1007/978-3-642-19805-2_8

[27] R. Atkey, N. Ghani, and P. Johann, "A relationally parametric model of dependent type theory," in *The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '14, San Diego, CA, USA, January 20-21, 2014.*, 2014.

[28] P. Martin-Löf, "An intuitionistic theory of types, predicative part," in *Logic Colloquium*. North Holland, 1975, pp. 73–118.

[29] P. Martin-Löf, *Intuitionistic type theory*, ser. Studies in proof theory. Bibliopolis, 1984, vol. 1.

[30] The Agda Team, *Agda User Manual, Release 2.6.4*, Jan 2023. [Online]. Available: https://readthedocs.org/projects/agda/downloads/pdf/latest

[31] The Coq Development Team, *The Coq Reference Manual, version 8.16.1*, Oct 2022. [Online]. Available: https://coq.inria.fr/distrib/current/refman/

[32] L. M. de Moura, S. Kong, J. Avigad, F. van Doorn, and J. von Raumer, "The lean theorem prover (system description)," in *Automated Deduction - CADE-25 - 25th International Conference on Automated Deduction, Berlin, Germany, August 1-7, 2015, Proceedings*, ser. Lecture Notes in Computer Science, A. P. Felty and A. Middeldorp, Eds., vol. 9195. Springer, 2015, pp. 378–388. [Online]. Available: https://doi.org/10.1007/978-3-319-21401-6_26

[33] K. Kapulkin and P. L. Lumsdaine, "The simplicial model of univalent foundations (after voevodsky)," *Journal of the European Mathematical Society*, vol. 23, no. 6, pp. 2071–2126, 2021.

[34] The Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study, 2013. [Online]. Available: https://homotopytypetheory.org/book/

[35] N. Kraus, "Internal ∞-categorical models of dependent type theory : Towards 2LTT eating HoTT," in *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*. IEEE, 2021, pp. 1–14. [Online]. Available: https://doi.org/10.1109/LICS52264.2021.9470667

[36] G. Gilbert, J. Cockx, M. Sozeau, and N. Tabareau, "Definitional proof-irrelevance without K," *Proc. ACM Program. Lang.*, vol. 3, no. POPL, pp. 3:1–3:28, 2019. [Online]. Available: https://doi.org/10.1145/3290316

[37] H. Herbelin, "Investigations into syntactic iterated parametricity and cubical type theory," 2020. [Online]. Available: https://hott-uf.github.io/2020/HoTTUF_2020_paper_22.pdf