Cubical Type Theory: a constructive interpretation of the univalence axiom*

Cyril Cohen[1], Thierry Coquand[2], Simon Huber[2], and Anders Mörtberg[3]

1 **Team MARELLE**
Inria Sophia Antipolis – Méditerranée, Sophia Antipolis, France
cyril.cohen@inria.fr

2 **Department of Computer Science and Engineering**
University of Gothenburg, Gothenburg, Sweden
thierry.coquand@cse.gu.se
simon.huber@cse.gu.se

3 **School of Mathematics**
Institute for Advanced Study, Princeton, NJ, USA
amortberg@math.ias.edu

Abstract

This paper presents a type theory in which it is possible to directly manipulate n-dimensional cubes (points, lines, squares, cubes, etc.) based on an interpretation of dependent type theory in a cubical set model. This enables new ways to reason about identity types, for instance, function extensionality is directly provable in the system. Further, Voevodsky's univalence axiom is provable in this system. We also explain an extension with some higher inductive types like the circle and propositional truncation. Finally we provide semantics for this cubical type theory in a constructive meta-theory.

1998 ACM Subject Classification F.3.2 Semantics of Programming Languages, F.4.1 Mathematical Logic

Keywords and phrases Univalence Axiom, Dependent Type Theory, Cubical Sets

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

This work is a continuation of the program started in [6, 13] to provide a constructive justification of Voevodsky's univalence axiom [25]. This axiom allows many improvements for the formalization of mathematics in type theory: function extensionality, identification of isomorphic structures, etc. In order to preserve the good computational properties of type theory it is crucial that postulated constants have a computational interpretation. Like in [6, 13, 20] our work is based on a nominal extension of λ-calculus, using *names* to represent formally elements of the unit interval $[0, 1]$. This paper presents two main contributions.

The first one is a refinement of the semantics presented in [6, 13]. We add new operations on names corresponding to the fact that the interval $[0, 1]$ is canonically a de Morgan

* This material is based upon work supported by the National Science Foundation under agreement No. DMS-1128155. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

algebra [3]. This allows us to significantly simplify our semantical justifications. In the previous work, we noticed that it is crucial for the semantics of higher inductive types [24] to have a "diagonal" operation. By adding this operation we can provide a semantical justification of some higher inductive types and we give two examples (the spheres and propositional truncation). Another shortcoming of the previous work was that using path types as equality types did not provide a justification of the computation rule of the Martin-Löf identity type [17] as a judgmental equality. This problem has been solved by Andrew Swan [23], in the framework of [6, 13, 20], who showed that we can define a new type, *equivalent to*, but not judgmentally equal to the path type. This has a simple definition in the present framework.

The second contribution is the design of a type system[1] inspired by this semantics which extends Martin-Löf type theory [18, 17]. We add two new operations on contexts: addition of new names representing dimensions and a restriction operation. Using these we can define a notion of connectedness which generalizes the notion of being path-connected, and then a Kan composition operation that expresses that being connected is preserved along paths. We also define a new operation on types which expresses that this notion of connectedness is preserved by equivalences. The axiom of univalence, and composition for the universe, are then both expressible using this new operation.

The paper is organized as follows. The first part, Sections 2 to 7, presents the type system. The second part, Section 8, provides its semantics in cubical sets. Finally, in Section 9, we present two possible extensions: the addition of an identity type, and two examples of higher inductive types.

2 Basic type theory

In this section we introduce the version of dependent type theory on which the rest of the paper is based. This presentation is standard, but included for completeness. The type theory that we consider has a type of natural numbers, but no universes (we consider the addition of universes in Section 7). It also has β and η-conversion for dependent functions and surjective pairing for dependent pairs.

The syntax of contexts, terms and types is specified by:

$$
\begin{array}{llll}
\Gamma, \Delta & ::= & () \mid \Gamma, x : A & \text{Contexts} \\[1em]
t, u, A, B & ::= & x \mid \lambda x : A.\, t \mid t\ u \mid (x : A) \to B & \Pi\text{-types} \\
& \mid & (t, u) \mid t.1 \mid t.2 \mid (x : A) \times B & \Sigma\text{-types} \\
& \mid & \mathsf{0} \mid \mathsf{s}\ u \mid \mathsf{natrec}\ t\ u \mid \mathsf{N} & \text{Natural numbers}
\end{array}
$$

We write $A \to B$ for the non-dependent function space and $A \times B$ for the type of non-dependent pairs. Terms and types are considered up to α-equivalence of bound variables. Substitutions, written $\sigma = (x_1/u_1, \ldots, x_n/u_n)$, are defined to act on expressions as usual, i.e., simultaneously replacing x_i by u_i, renaming bound variables whenever necessary. The inference rules of this system are presented in Figure 1.

We define $\Delta \vdash \sigma : \Gamma$ by induction on Γ. We have $\Delta \vdash () : ()$ (empty substitution) and $\Delta \vdash (\sigma, x/u) : \Gamma, x : A$ if $\Delta \vdash \sigma : \Gamma$ and $\Delta \vdash u : A\sigma$.

[1] We have implemented a type-checker for this system in Haskell, which is available at: `https://github.com/mortberg/cubicaltt`

We write J for an arbitrary judgment and, as usual, we consider also *hypothetical* judgments $\Gamma \vdash \mathsf{J}$ in a *context* Γ.

The following lemma will be valid for all extensions of type theory we consider below.

▶ **Lemma 1.** *The following rules are admissible:*

1. *Weakening rules: a judgment valid in a context stays valid in any extension of this context.*
2. *Substitution rules:*

$$\frac{\Gamma \vdash \mathsf{J} \qquad \Delta \vdash \sigma : \Gamma}{\Delta \vdash \mathsf{J}\sigma}$$

3. *From* $\Gamma \vdash A = B$ *we can derive* $\Gamma \vdash A$ *and* $\Gamma \vdash B$*, and from* $\Gamma \vdash a = b : A$ *we can derive* $\Gamma \vdash a : A$ *and* $\Gamma \vdash b : A$*.*

3 Path types

As in [6, 20] we assume that we are given a discrete infinite set of names (representing directions) $i, j, k, \ldots$ We define $\mathbb{I}$ to be the free de Morgan algebra [3] on this set of names. The elements of $\mathbb{I}$ can be described by the following grammar:

$$r, s \quad ::= \quad 0 \mid 1 \mid i \mid 1 - r \mid r \wedge s \mid r \vee s$$

The set $\mathbb{I}$ also has decidable equality, and as a distributive lattice, it can be described as the free distributive lattice generated by symbols i and $1 - i$ [3]. As in [6], the elements in $\mathbb{I}$ can be thought as formal representations of elements in $[0, 1]$, with $r \wedge s$ representing $min(r, s)$ and $r \vee s$ representing $max(r, s)$.

Contexts can now be extended with name declarations:

$$\Gamma, \Delta \quad ::= \quad \ldots \mid \Gamma, i : \mathbb{I}$$

together with the context rule:

$$\frac{\Gamma \vdash}{\Gamma, i : \mathbb{I} \vdash} \; (i \notin \mathrm{dom}(\Gamma))$$

A judgment of the form $\Gamma \vdash r : \mathbb{I}$ means that $\Gamma \vdash$ and r is an element of $\mathbb{I}$ depending only on the names declared in Γ. The judgment $\Gamma \vdash r = s : \mathbb{I}$ means that r and s are equal as elements of $\mathbb{I}$, $\Gamma \vdash r : \mathbb{I}$, and $\Gamma \vdash s : \mathbb{I}$. Note, that judgmental equality for $\mathbb{I}$ will be re-defined once we introduce restricted contexts in Section 4.

3.1 Syntax and inference rules

The extension to the syntax of basic dependent type theory is:

$$\begin{array}{lcll} t, u, A, B & ::= & \ldots & \\ & \mid & \mathsf{Path}\ A\ t\ u \mid \langle i \rangle\ t \mid t\ r & \qquad\qquad \text{Path types} \end{array}$$

Path abstraction, $\langle i \rangle\ t$, binds the name i in t, and path application, $t\ r$, applies a term t to an element $r : \mathbb{I}$. This is similar to the notion of name-abstraction in nominal sets [19].

The substitution operation now has to be extended to substitutions of the form (i/r). There are special substitutions of the form $(i/0)$ and $(i/1)$ corresponding to taking faces of an n-dimensional cube, we write these simply as $(i0)$ and $(i1)$.

$\boxed{\text{Well-formed contexts, } \Gamma \vdash}$ (The condition $x \notin \mathrm{dom}(\Gamma)$ means that x is not declared in Γ)

$$\frac{}{() \vdash} \qquad \frac{\Gamma \vdash A}{\Gamma, x : A \vdash}\ (x \notin \mathrm{dom}(\Gamma))$$

$\boxed{\text{Well-formed types, } \Gamma \vdash A}$

$$\frac{\Gamma, x : A \vdash B}{\Gamma \vdash (x : A) \to B} \qquad \frac{\Gamma, x : A \vdash B}{\Gamma \vdash (x : A) \times B} \qquad \frac{\Gamma \vdash}{\Gamma \vdash \mathsf{N}}$$

$\boxed{\text{Well-typed terms, } \Gamma \vdash t : A}$

$$\frac{\Gamma \vdash t : A \qquad \Gamma \vdash A = B}{\Gamma \vdash t : B} \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x : A.\, t : (x : A) \to B} \qquad \frac{\Gamma \vdash}{\Gamma \vdash x : A}\ (x : A \in \Gamma)$$

$$\frac{\Gamma \vdash t : (x : A) \to B \qquad \Gamma \vdash u : A}{\Gamma \vdash t\ u : B(x/u)} \qquad \frac{\Gamma \vdash t : (x : A) \times B}{\Gamma \vdash t.1 : A} \qquad \frac{\Gamma \vdash t : (x : A) \times B}{\Gamma \vdash t.2 : B(x/t.1)}$$

$$\frac{\Gamma \vdash t : A \qquad \Gamma \vdash u : B(x/t)}{\Gamma \vdash (t, u) : (x : A) \times B} \qquad \frac{\Gamma \vdash}{\Gamma \vdash 0 : \mathsf{N}} \qquad \frac{\Gamma \vdash n : \mathsf{N}}{\Gamma \vdash \mathsf{s}\ n : \mathsf{N}}$$

$$\frac{\Gamma, x : N \vdash P \qquad \Gamma \vdash a : P(x/0) \qquad \Gamma \vdash b : (n : \mathsf{N}) \to P(x/n) \to P(x/\mathsf{s}\ n)}{\Gamma \vdash \mathsf{natrec}\ a\ b : (x : \mathsf{N}) \to P}$$

$\boxed{\text{Type equality, } \Gamma \vdash A = B}$ (Congruence and equivalence rules which are omitted)

$\boxed{\text{Term equality, } \Gamma \vdash a = b : A}$ (Congruence and equivalence rules are omitted)

$$\frac{\Gamma \vdash t = u : A \qquad \Gamma \vdash A = B}{\Gamma \vdash t = u : B}$$

$$\frac{\Gamma, x : A \vdash t : B \qquad \Gamma \vdash u : A}{\Gamma \vdash (\lambda x : A.\, t)\ u = t(x/u) : B(x/u)} \qquad \frac{\Gamma, x : A \vdash t\ x = u\ x : B}{\Gamma \vdash t = u : (x : A) \to B}$$

$$\frac{\Gamma \vdash t : A \qquad \Gamma \vdash u : B(x/t)}{\Gamma \vdash (t, u).1 = t : A} \qquad \frac{\Gamma \vdash t : A \qquad \Gamma \vdash u : B(x/t)}{\Gamma \vdash (t, u).2 = u : B(x/t)}$$

$$\frac{\Gamma \vdash t.1 = u.1 : A \qquad \Gamma \vdash t.2 = u.2 : B(x/t.1)}{\Gamma \vdash t = u : (x : A) \times B}$$

$$\frac{\Gamma, x : \mathsf{N} \vdash P \qquad \Gamma \vdash a : P(x/0) \qquad \Gamma \vdash b : (n : \mathsf{N}) \to P(x/n) \to P(x/\mathsf{s}\ n)}{\Gamma \vdash \mathsf{natrec}\ a\ b\ 0 = a : P(x/0)}$$

$$\frac{\Gamma, x : \mathsf{N} \vdash P \qquad \Gamma \vdash a : P(x/0) \qquad \Gamma \vdash b : (n : \mathsf{N}) \to P(x/n) \to P(x/\mathsf{s}\ n) \qquad \Gamma \vdash n : \mathsf{N}}{\Gamma \vdash \mathsf{natrec}\ a\ b\ (\mathsf{s}\ n) = b\ n\ (\mathsf{natrec}\ a\ b\ n) : P(x/\mathsf{s}\ n)}$$

Figure 1 Inference rules of the basic type theory

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash t : A \qquad \Gamma \vdash u : A}{\Gamma \vdash \mathsf{Path}\ A\ t\ u} \qquad \frac{\Gamma \vdash A \qquad \Gamma, i : \mathbb{I} \vdash t : A}{\Gamma \vdash \langle i \rangle\ t : \mathsf{Path}\ A\ t(i0)\ t(i1)}$$

$$\frac{\Gamma \vdash t : \mathsf{Path}\ A\ u_0\ u_1 \qquad \Gamma \vdash r : \mathbb{I}}{\Gamma \vdash t\ r : A} \qquad \frac{\Gamma \vdash A \qquad \Gamma, i : \mathbb{I} \vdash t : A}{\Gamma \vdash (\langle i \rangle\ t)\ r = t(i/r) : A}$$

$$\frac{\Gamma, i : \mathbb{I} \vdash t\ i = u\ i : A}{\Gamma \vdash t = u : \mathsf{Path}\ A\ u_0\ u_1} \qquad \frac{\Gamma \vdash t : \mathsf{Path}\ A\ u_0\ u_1}{\Gamma \vdash t\ 0 = u_0 : A} \qquad \frac{\Gamma \vdash t : \mathsf{Path}\ A\ u_0\ u_1}{\Gamma \vdash t\ 1 = u_1 : A}$$

Figure 2 Inference rules for path types

The inference rules for path types are presented in Figure 2.

We define $1_a : \mathsf{Path}\ A\ a\ a$ as $1_a = \langle i \rangle\ a$, which corresponds to a proof of reflexivity.

The intuition is that a type in a context with n names corresponds to an n-dimensional cube:

$() \vdash A$	$\bullet\ A$
$i : \mathbb{I} \vdash A$	$A(i0) \xrightarrow{A} A(i1)$
$i : \mathbb{I}, j : \mathbb{I} \vdash A$	$\begin{tikzcd} A(i0)(j1) \arrow[r, "A(j1)"] & A(i1)(j1) \\ A(i0)(j0) \arrow[u, "A(i0)"] \arrow[r, "A(j0)"'] \arrow[ur, phantom, "A"] & A(i1)(j0) \arrow[u, "A(i1)"'] \end{tikzcd}$
$\vdots$	$\vdots$

Note that $A(i0)(j0) = A(j0)(i0)$. The substitution (i/j) corresponds to renaming a dimension, while $(i/1-i)$ corresponds to the inversion of a path. If we have:

$$a \xrightarrow{p} b$$

in direction i, then:

$$b \xrightarrow{p(i/1-i)} a$$

The substitutions $(i/i \wedge j)$ and $(i/i \vee j)$ correspond to special kinds of degeneracies called *connections* [7]. The connections $p(i/i \wedge j)$ and $p(i/i \vee j)$ can be drawn as the squares:

$$\begin{tikzcd} a \arrow[r, "p"] & b \\ a \arrow[u, "p(i0)"] \arrow[r, "p(i0)"'] \arrow[ur, phantom, "p(i/i \wedge j)"] & a \arrow[u, "p(i/j)"'] \end{tikzcd} \qquad \begin{tikzcd} b \arrow[r, "p(i1)"] & b \\ a \arrow[u, "p(i/j)"] \arrow[r, "p"'] \arrow[ur, phantom, "p(i/i \vee j)"] & b \arrow[u, "p(i1)"'] \end{tikzcd} \qquad \begin{tikzcd} j \\ \cdot \arrow[u] \arrow[r] & i \end{tikzcd}$$

where, for instance, the right hand side of the left square is computed as

$$p(i/i \wedge j)(i1) \quad = \quad p(i/1 \wedge j) \quad = \quad p(i/j)$$

and the bottom and left hand sides are degenerate.

3.2 Examples

Representing equalities using path types allows novel definitions of many standard operations on identity types that are usually proved by identity elimination. For instance, the fact that the images of two equal elements are equal can be defined as:

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : A \qquad \Gamma \vdash f : A \to B \qquad \Gamma \vdash p : \mathsf{Path}\ A\ a\ b}{\Gamma \vdash \langle i \rangle\ f\ (p\ i) : \mathsf{Path}\ B\ (f\ a)\ (f\ b)}$$

This operation satisfies some judgmental equalities that do not hold judgmentally when the identity type is defined as an inductive family (see Section 7.2 of [6] for details).

We can also define new operations, for instance, function extensionality for path types can be proved as:

$$\frac{\Gamma \vdash f : (x : A) \to B \qquad \Gamma \vdash g : (x : A) \to B \qquad \Gamma \vdash p : (x : A) \to \mathsf{Path}\ B\ (f\ x)\ (g\ x)}{\Gamma \vdash \langle i \rangle\ \lambda x : A.\ p\ x\ i : \mathsf{Path}\ ((x : A) \to B)\ f\ g}$$

To see that this is correct we check that the term has the correct faces, for instance:

$$(\langle i \rangle\ \lambda x : A.\ p\ x\ i)\ 0 \ = \ \lambda x : A.\ p\ x\ 0 \ = \ \lambda x : A.\ f\ x \ = \ f$$

We can also justify the fact that singletons are contractible, that is, that any elements in $(x : A) \times (\mathsf{Path}\ A\ a\ x)$ is equal to $(a, 1_a)$:

$$\frac{\Gamma \vdash p : \mathsf{Path}\ A\ a\ b}{\Gamma \vdash \langle i \rangle\ (p\ i, \langle j \rangle\ p\ (i \wedge j)) : \mathsf{Path}\ ((x : A) \times (\mathsf{Path}\ A\ a\ x))\ (a, 1_a)\ (b, p)}$$

As in the previous work [6, 13] we need to add *composition operations* that are defined by induction on the type.

4 Systems, composition, and transport

In this section we define the operation of context *restriction* which will allow us to describe new geometrical shapes corresponding to "sub-polyhedra" of a cube. Using this we can define the composition operation. From this operation we will also be able to define the transport operation and the elimination principle for Path types.

4.1 The face lattice

The *face lattice*, $\mathbb{F}$, is the free distributive lattice on symbols $(i = 0)$ and $(i = 1)$ with the relation $(i = 0) \wedge (i = 1) = 0_{\mathbb{F}}$. The elements of the face lattice can be described by the syntax

$$\varphi, \psi \quad ::= \quad 0_{\mathbb{F}} \mid 1_{\mathbb{F}} \mid (i = 0) \mid (i = 1) \mid \varphi \wedge \psi \mid \varphi \vee \psi$$

There is a canonical lattice map $\mathbb{I} \to \mathbb{F}$ sending i to $(i = 1)$ and $1 - i$ to $(i = 0)$. We write $(r = 1)$ for the image of $r : \mathbb{I}$ in $\mathbb{F}$ and we write $(r = 0)$ for $(1 - r = 1)$. We have $(r = 1) \wedge (r = 0) = 0_{\mathbb{F}}$ and we define the lattice map $\mathbb{F} \to \mathbb{F}$, $\psi \longmapsto \psi(i/r)$ sending $(i = 1)$ to $(r = 1)$ and $(i = 0)$ to $(r = 0)$.

Any element of $\mathbb{F}$ is the join of the irreducible elements below it. An irreducible element of this lattice is a *face*, i.e., a conjunction of elements of the form $(i = 0)$ and $(j = 1)$.

This provides a disjunctive normal form for elements of $\mathbb{F}$, and it follows from this that the equality on $\mathbb{F}$ is decidable.

Geometrically, the elements of $\mathbb{F}$ describe "sub-polyhedra" of a cube. For instance, the element $(i = 0) \vee (j = 1)$ can be seen as the union of two faces of the square in directions j and i. If I is a finite set of names, we define the *boundary* of I as the element ∂_I of $\mathbb{F}$ which is the disjunction of all $(i = 0) \vee (i = 1)$ for i in I. It is the greatest element depending at most on elements in I which is $< 1_{\mathbb{F}}$.

We write $\Gamma \vdash \psi : \mathbb{F}$ to mean that ψ is an element of $\mathbb{F}$ using only the names declared in Γ. We introduce then the new *restriction* operation on contexts:

$$\Gamma, \Delta \quad ::= \quad \ldots \mid \Gamma, \varphi$$

together with the rule:

$$\frac{\Gamma \vdash \varphi : \mathbb{F}}{\Gamma, \varphi \vdash}$$

This allows us to describe new geometrical shapes: as we have seen above, a type in a context $\Gamma = i : \mathbb{I}, j : \mathbb{I}$ can be thought of as a square, and a type in the restricted context Γ, φ will then represent a compatible union of faces of this square. This can be illustrated by:

$i : \mathbb{I}, (i = 0) \vee (i = 1) \vdash A$	$A(i0) \bullet \qquad \bullet A(i1)$
$i : \mathbb{I}, j : \mathbb{I}, (i = 0) \vee (j = 1) \vdash A$	$\begin{tikzcd} A(i0)(j1) \arrow[r, "A(j1)"] & A(i1)(j1) \\ A(i0)(j0) \arrow[u, "A(i0)"] & \end{tikzcd}$
$i : \mathbb{I}, j : \mathbb{I}, (i = 0) \vee (i = 1) \vee (j = 0) \vdash A$	$\begin{tikzcd} A(i0)(j1) & A(i1)(j1) \\ A(i0)(j0) \arrow[u, "A(i0)"] \arrow[r, "A(j0)"'] & A(i1)(j0) \arrow[u, "A(i1)"'] \end{tikzcd}$

There is a canonical map from the lattice $\mathbb{F}$ to the congruence lattice of $\mathbb{I}$, which is distributive [3], sending $(i = 1)$ to the congruence identifying i with 1 (and $1 - i$ with 0) and sending $(i = 0)$ to the congruence identifying i with 0 (and $1 - i$ with 1). In this way, any element ψ of $\mathbb{F}$ defines a congruence $r = s \ (\mathsf{mod.}\ \psi)$ on $\mathbb{I}$.

This congruence can be described as a substitution if ψ is irreducible; for instance, if ψ is $(i = 0) \wedge (j = 1)$ then $r = s \ (\mathsf{mod.}\ \psi)$ is equivalent to $r(i0)(j1) = s(i0)(j1)$. The congruence associated to $\psi = \varphi_0 \vee \varphi_1$ is the meet of the congruences associated to φ_0 and φ_1 respectively, so that we have, e.g., $i = 1 - j \ (\mathsf{mod.}\ \psi)$ if $\varphi_0 = (i = 0) \wedge (j = 1)$ and $\varphi_1 = (i = 1) \wedge (j = 0)$.

To any context Γ we can associate recursively a congruence on $\mathbb{I}$, the congruence on Γ, ψ being the join of the congruence defined by Γ and the congruence defined by ψ. The congruence defined by $()$ is equality in $\mathbb{I}$, and an extension $x : A$ or $i : \mathbb{I}$ does not change the congruence. The judgment $\Gamma \vdash r = s : \mathbb{I}$ then means that $r = s \ (\mathsf{mod.}\ \Gamma)$.

In the case where Γ does not use the restriction operation, this judgment means $r = s$ in $\mathbb{I}$. If i is declared in Γ, then $\Gamma, (i = 0) \vdash r = s : \mathbb{I}$ is equivalent to $\Gamma \vdash r(i0) = s(i0) : \mathbb{I}$. Similarly any context Γ defines a congruence on $\mathbb{F}$ with $\Gamma, \psi \vdash \varphi_0 = \varphi_1 : \mathbb{F}$ being equivalent to $\Gamma \vdash \psi \wedge \varphi_0 = \psi \wedge \varphi_1 : \mathbb{F}$.

As explained above, the elements of $\mathbb{I}$ can be seen as formal representations of elements in the interval $[0, 1]$. The elements of $\mathbb{F}$ can then be seen as formulas on elements of $[0, 1]$. We have a simple form of *quantifier elimination* on $\mathbb{F}$: given an element φ of $\mathbb{F}$ and a name i, we can define $\forall i.\varphi$ as being the disjunction of all irreducible elements $\leqslant \varphi$ independent of i. If ψ is independent of i, we then have $\psi \leqslant \varphi$ if and only if $\psi \leqslant \forall i.\varphi$. For example, if φ is $(i = 0) \vee ((i = 1) \wedge (j = 0)) \vee (j = 1)$ then $\forall i.\varphi$ is $(j = 1)$. This operation will play a crucial role in Section 6.2 for the definition of composition of glueing.

Since $\mathbb{F}$ is not a Boolean algebra, we don't have in general $\varphi = (\varphi \wedge (i = 0)) \vee (\varphi \wedge (i = 1))$, but we always have the following decomposition:

▶ **Lemma 2.** *For any element φ of $\mathbb{F}$ and any name i we have*

$$\varphi = (\forall i.\varphi) \vee (\varphi \wedge (i = 0)) \vee (\varphi \wedge (i = 1))$$

We also have $\varphi \wedge (i = 0) \leqslant \varphi(i0)$ and $\varphi \wedge (i = 1) \leqslant \varphi(i1)$.

4.2 Syntax and inference rules for systems

The extension to the syntax of dependent type theory with path types is:

$$\begin{array}{lcll} t, u, A, B & ::= & \dots & \\ & | & S & \\ S & ::= & [\, \varphi_1\ t_1, \dots, \varphi_n\ t_n \,] & \qquad\qquad \text{Systems} \end{array}$$

We allow $n = 0$ and get the empty system $[\,]$. As explained above, a context now corresponds in general to the union of sub-faces of a cube. In Figure 3 we provide operations for combining compatible systems of types and elements, the side condition for these rules is that $\Gamma \vdash \varphi_1 \vee \dots \vee \varphi_n = 1_{\mathbb{F}} : \mathbb{F}$. This condition requires Γ to be sufficiently restricted: for example $\Delta, (i = 0) \vee (i = 1) \vdash (i = 0) \vee (i = 1) = 1_{\mathbb{F}}$.

$$\frac{\Gamma, \varphi_1 \vdash A_1 \qquad \cdots \qquad \Gamma, \varphi_n \vdash A_n \qquad \Gamma, \varphi_i \wedge \varphi_j \vdash A_i = A_j}{\Gamma \vdash [\, \varphi_1\ A_1, \dots, \varphi_n\ A_n \,]}$$

$$\frac{\Gamma \vdash A \qquad \Gamma, \varphi_1 \vdash t_1 : A \qquad \cdots \qquad \Gamma, \varphi_n \vdash t_n : A \qquad \Gamma, \varphi_i \wedge \varphi_j \vdash t_i = t_j : A}{\Gamma \vdash [\, \varphi_1\ t_1, \dots, \varphi_n\ t_n \,] : A}$$

$$\frac{\Gamma, \varphi_1 \vdash \mathsf{J} \qquad \cdots \qquad \Gamma, \varphi_n \vdash \mathsf{J}}{\Gamma \vdash \mathsf{J}} \qquad\qquad \frac{\Gamma \vdash \varphi_i = 1_{\mathbb{F}} : \mathbb{F}}{\Gamma \vdash [\, \varphi_1\ A_1, \dots, \varphi_n\ A_n \,] = A_i}$$

$$\frac{\Gamma \vdash [\, \varphi_1\ t_1, \dots, \varphi_n\ t_n \,] : A \qquad \Gamma \vdash \varphi_i = 1_{\mathbb{F}} : \mathbb{F}}{\Gamma \vdash [\, \varphi_1\ t_1, \dots, \varphi_n\ t_n \,] = t_i : A}$$

Figure 3 Inference rules for systems

Note that when $n = 0$ the second of the above rules should be read as: if $\Gamma \vdash 0_{\mathbb{F}} = 1_{\mathbb{F}} : \mathbb{F}$ and $\Gamma \vdash A$, then $\Gamma \vdash [\,] : A$.

We extend the definition of the substitution judgment by $\Delta \vdash \sigma : \Gamma, \varphi$ if $\Delta \vdash \sigma : \Gamma$, $\Gamma \vdash \varphi : \mathbb{F}$, and $\Delta \vdash \varphi\sigma = 1_{\mathbb{F}} : \mathbb{F}$.

If $\Gamma, \varphi \vdash u : A$, then $\Gamma \vdash a : A[\varphi \mapsto u]$ is an abbreviation for $\Gamma \vdash a : A$ and $\Gamma, \varphi \vdash a = u : A$. In this case, we see this element a as a witness that the partial element u, defined on the "extent" φ (using the terminology from [10]), is *connected*. More generally, we write $\Gamma \vdash a : A[\varphi_1 \mapsto u_1, \dots, \varphi_k \mapsto u_k]$ for $\Gamma \vdash a : A$ and $\Gamma, \varphi_i \vdash a = u_i : A$ for $i = 1, \dots, k$.

For instance, if $\Gamma, i : \mathbb{I} \vdash A$ and $\Gamma, i : \mathbb{I}, \varphi \vdash u : A$ where $\varphi = (i = 0) \vee (i = 1)$ then the element u is determined by two elements $\Gamma \vdash a_0 : A(i0)$ and $\Gamma \vdash a_1 : A(i1)$ and an element $\Gamma, i : \mathbb{I} \vdash a : A[(i = 0) \mapsto a_0, (i = 1) \mapsto a_1]$ gives a path connecting a_0 and a_1.

▶ **Lemma 3.** *The following rules are admissible:*[2]

$$\frac{\Gamma \vdash \varphi \leqslant \psi \qquad \Gamma, \psi \vdash \mathsf{J}}{\Gamma, \varphi \vdash \mathsf{J}} \qquad\qquad \frac{\Gamma, 1_{\mathbb{F}} \vdash \mathsf{J}}{\Gamma \vdash \mathsf{J}} \qquad\qquad \frac{\Gamma, \varphi, \psi \vdash \mathsf{J}}{\Gamma, \varphi \wedge \psi \vdash \mathsf{J}}$$

Furthermore, if φ is independent of i, the following rules are admissible

$$\frac{\Gamma, i : \mathbb{I}, \varphi \vdash \mathsf{J}}{\Gamma, \varphi, i : \mathbb{I} \vdash \mathsf{J}}$$

and it follows that we have in general:

$$\frac{\Gamma, i : \mathbb{I}, \varphi \vdash \mathsf{J}}{\Gamma, \forall i.\varphi, i : \mathbb{I} \vdash \mathsf{J}}$$

4.3 Composition operation

The syntax of compositions is given by:

$$\begin{array}{lcll} t, u, A, B & ::= & \dots & \\ & \mid & \mathsf{comp}^i\ A\ [\varphi \mapsto u]\ a_0 & \text{Compositions} \end{array}$$

where u is a system on the extent φ.

The composition operation expresses that being connected is preserved along paths: if a partial path is connected at 0, then it is connected at 1.

$$\frac{\Gamma \vdash \varphi \qquad \Gamma, i : \mathbb{I} \vdash A \qquad \Gamma, \varphi, i : \mathbb{I} \vdash u : A \qquad \Gamma \vdash a_0 : A(i0)[\varphi \mapsto u(i0)]}{\Gamma \vdash \mathsf{comp}^i\ A\ [\varphi \mapsto u]\ a_0 : A(i1)[\varphi \mapsto u(i1)]}$$

Note that comp^i binds i in A and u and that we have in particular the following equality judgments for systems:

$$\Gamma \vdash \mathsf{comp}^i\ A\ [1_{\mathbb{F}} \mapsto u]\ a_0 = u(i1)$$

If we have a substitution $\Delta \vdash \sigma : \Gamma$, then

$$(\mathsf{comp}^i\ A\ [\varphi \mapsto u]\ a_0)\sigma = \mathsf{comp}^j\ A(\sigma, i/j)\ [\varphi\sigma \mapsto u(\sigma, i/j)]\ a_0\sigma$$

where j is fresh for Δ, which corresponds semantically to the *uniformity* [6, 13] of the composition operation.

We use the abbreviation $[\varphi_1 \mapsto u_1, \dots, \varphi_n \mapsto u_n]$ for $[\bigvee_i \varphi_i \mapsto [\varphi_1\ u_1, \dots, \varphi_n\ u_n]]$ and in particular we write $[]$ for $[0_{\mathbb{F}} \mapsto [\,]]$.

[2] The inference rules with double line are each a pair of rules, because they can be used in both directions.

▶ **Example 4.** With composition we can justify transitivity of path types:

$$\frac{\Gamma \vdash p : \mathsf{Path}\ A\ a\ b \qquad \Gamma \vdash q : \mathsf{Path}\ A\ b\ c}{\Gamma \vdash \langle i \rangle\ \mathsf{comp}^j\ A\ [(i=0) \mapsto a, (i=1) \mapsto q\ j]\ (p\ i) : \mathsf{Path}\ A\ a\ c}$$

This composition can be visualized as the dashed arrow in the square:

$$\begin{tikzcd}
a \arrow[r, dashed] & c \\
a \arrow[u, "a"] \arrow[r, "p\ i"'] & b \arrow[u, "q\ j"']
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
{} & \\
\cdot \arrow[u, "j"] \arrow[r, "i"'] & {}
\end{tikzcd}$$

4.4 Kan filling operation

As we have connections we also get Kan filling operations from compositions:

$$\Gamma, i : \mathbb{I} \vdash \mathsf{fill}^i\ A\ [\varphi \mapsto u]\ a_0 = \mathsf{comp}^j\ A(i/i \wedge j)\ [\varphi \mapsto u(i/i \wedge j), (i=0) \mapsto a_0]\ a_0 : A$$

where j is fresh for Γ. The element $\Gamma, i : \mathbb{I} \vdash v = \mathsf{fill}^i\ A\ [\varphi \mapsto u]\ a_0 : A$ satisfies:

$$\Gamma \vdash v(i0) = a_0 : A(i0) \quad \Gamma \vdash v(i1) = \mathsf{comp}^i\ A\ [\varphi \mapsto u]\ a_0 : A(i1) \quad \Gamma, \varphi \vdash v(i1) = u : A$$

This means that we can not only compute the lid of an open box but also its filling. If φ is the boundary formula on the names declared in Γ, we recover the Kan operation for cubical sets [14].

4.5 Equality judgments for composition

The equality judgments for $\mathsf{comp}^i\ C\ [\varphi \mapsto u]\ a_0$ are defined by cases on the type C. There are five cases to consider:

Product types

In the case of a product type $\Gamma, i : \mathbb{I} \vdash C = (x : A) \to B$, we have $\Gamma, \varphi, i : \mathbb{I} \vdash \mu : C$ and $\Gamma \vdash \lambda_0 : C(i0)[\varphi \mapsto \mu(i0)]$. As the composition will be of type $C(i1)$ we have to explain how it behaves when applied to some $\Gamma \vdash u_1 : A(i1)$

$$\Gamma \vdash (\mathsf{comp}^i\ C\ [\varphi \mapsto \mu]\ \lambda_0)\ u_1 = \mathsf{comp}^i\ B(x/v)\ [\varphi \mapsto \mu\ v]\ (\lambda_0\ v(i0)) : B(i1)[\varphi \mapsto (\mu\ v)(i1)]$$

where

$$\Gamma, i : \mathbb{I} \vdash w = \mathsf{fill}^i\ A(i/1-i)\ []\ u_1 : A(i/1-i) \quad \text{and} \quad \Gamma, i : \mathbb{I} \vdash v = w(i/1-i) : A.$$

Sum types

In the case of a sum type $\Gamma, i : \mathbb{I} \vdash C = (x : A) \times B$, we have $\Gamma, \varphi, i : \mathbb{I} \vdash w : C$ and $\Gamma \vdash w_0 : C(i0)[\varphi \mapsto w(i0)]$. Let $\Gamma, i : \mathbb{I} \vdash a = \mathsf{fill}^i\ A\ [\varphi \mapsto w.1]\ w_0.1 : A$ and:

$$\Gamma \vdash c_1 = \mathsf{comp}^i\ A\ [\varphi \mapsto w.1]\ w_0.1 \qquad \Gamma \vdash c_2 = \mathsf{comp}^i\ B(x/a)\ [\varphi \mapsto w.2]\ w_0.2$$

Using this we define:

$$\Gamma \vdash \mathsf{comp}^i\ C\ [\varphi \mapsto w]\ w_0 = (c_1, c_2) : C(i1)[\varphi \mapsto w(i1)]$$

Natural numbers

In the case of $\Gamma, i : \mathbb{I} \vdash C = \mathsf{N}$ we define $\mathsf{comp}^i\ C\ [\varphi \mapsto n]\ n_0$ by recursion:

$$\Gamma \vdash \mathsf{comp}^i\ C\ [\varphi \mapsto \mathsf{0}]\ \mathsf{0} = \mathsf{0} : C[\varphi \mapsto \mathsf{0}]$$
$$\Gamma \vdash \mathsf{comp}^i\ C\ [\varphi \mapsto \mathsf{s}\ n]\ (\mathsf{s}\ n_0) = \mathsf{s}\ (\mathsf{comp}^i\ C\ [\varphi \mapsto n]\ n_0) : C[\varphi \mapsto \mathsf{s}\ n]$$

Path types

In the case of a path type $\Gamma, i : \mathbb{I} \vdash C = \mathsf{Path}\ A\ u\ v$, we have $\Gamma, \varphi, i : \mathbb{I} \vdash p : C$ and $\Gamma \vdash p_0 : C(i0)[\varphi \mapsto p(i0)]$. We define

$$\Gamma \vdash \mathsf{comp}^i\ C\ [\varphi \mapsto p]\ p_0 = \langle j \rangle\ \mathsf{comp}^i\ A\ S\ (p_0\ j) : C(i1)[\varphi \mapsto p(i1)]$$

where the system S is $[\varphi \mapsto p\ j, (j = 0) \mapsto u, (j = 1) \mapsto v]$.

4.6 Transport

Composition for $\varphi = 0_{\mathbb{F}}$ corresponds to transport:

$$\Gamma \vdash \mathsf{transp}^i\ A\ a = \mathsf{comp}^i\ A\ []\ a : A(i1)$$

Together with the fact that singletons are contractible, from Section 3.2, we get the elimination principle for Path types in the same manner as explained for identity types in Section 7.2 of [6].

5 Derived notions and operations

This section defines various notions and operations that will be used for defining compositions for the glue operation in the next section. This operation will then be used to define the composition operation for the universe and to prove the univalence axiom.

5.1 Contractible types

We define $\mathsf{isContr}\ A = (x : A) \times ((y : A) \to \mathsf{Path}\ A\ x\ y)$. A proof of $\mathsf{isContr}\ A$ witnesses the fact that A is *contractible*.

Given $\Gamma \vdash p : \mathsf{isContr}\ A$ and $\Gamma, \varphi \vdash u : A$ we define the operation[3]

$$\Gamma \vdash \mathsf{contr}\ p\ [\varphi \mapsto u] = \mathsf{comp}^i\ A\ [\varphi \mapsto p.2\ u\ i]\ p.1 : A[\varphi \mapsto u]$$

Conversely, we can state the following characterization of contractible types:

▶ **Lemma 5.** *Let $\Gamma \vdash A$ and assume that we have one operation*

$$\frac{\Gamma, \varphi \vdash u : A}{\Gamma \vdash \mathsf{contr}\ [\varphi \mapsto u] : A[\varphi \mapsto u]}$$

then we can find an element in $\mathsf{isContr}\ A$.

Proof. We define $x = \mathsf{contr}\ [] : A$ and prove that any element $y : A$ is path equal to x. For this, we introduce a fresh name $i : \mathbb{I}$ and define $\varphi = (i = 0) \vee (i = 1)$ and $u = [(i = 0) \mapsto x, (i = 1) \mapsto y]$. Using this we obtain $\Gamma, i : \mathbb{I} \vdash v = \mathsf{contr}\ [\varphi \mapsto u] : A[\varphi \mapsto u]$. In this way, we get a path $\langle i \rangle \mathsf{contr}\ [\varphi \mapsto u]$ connecting x and y. ◀

[3] This expresses that the restriction map $\Gamma, \varphi \to \Gamma$ has the left lifting property w.r.t. any "trivial fibration", i.e., contractible extensions $\Gamma, x : A \to \Gamma$. The restriction maps $\Gamma, \varphi \to \Gamma$ thus represent "cofibrations" while the maps $\Gamma, x : A \to \Gamma$ represent "fibrations".

5.2 The pres operation

The pres operation states that the image of a composition is path equal to the composition of the respective images, so that any function *preserves* composition, up to path equality.

▶ **Lemma 6.** *We have an operation:*

$$\frac{\Gamma, i : \mathbb{I} \vdash f : T \to A \qquad \Gamma \vdash \varphi \qquad \Gamma, \varphi, i : \mathbb{I} \vdash t : T \qquad \Gamma \vdash t_0 : T(i0)[\varphi \mapsto t(i0)]}{\Gamma \vdash \mathsf{pres}^i\ f\ [\varphi \mapsto t]\ t_0 : (\mathsf{Path}\ A(i1)\ c_1\ c_2)[\varphi \mapsto \langle j \rangle\ (f\ t)(i1)]}$$

where $c_1 = \mathsf{comp}^i\ A\ [\varphi \mapsto f\ t]\ (f(i0)\ t_0)$ *and* $c_2 = f(i1)\ (\mathsf{comp}^i\ T\ [\varphi \mapsto t]\ t_0)$.

Proof. Let $\Gamma \vdash a_0 = f(i0)\ t_0 : A(i0)$ and $\Gamma, \varphi, i : \mathbb{I} \vdash a = f\ t : A$, together with $\Gamma, i : \mathbb{I} \vdash v = \mathsf{fill}^i\ T\ [\varphi \mapsto t]\ t_0 : T$. We take $\mathsf{pres}^i\ f\ [\varphi \mapsto t]\ t_0 = \langle j \rangle\ \mathsf{comp}^i\ A\ [\varphi \mapsto a, (j = 1) \mapsto f\ v]\ a_0$. ◀

Note that pres^i binds i in f and t.

5.3 The equiv operation

We define $\mathsf{isEquiv}\ T\ A\ f = (y : A) \to \mathsf{isContr}\ ((x : T) \times \mathsf{Path}\ A\ y\ (f\ x))$ and $\mathsf{Equiv}\ T\ A = (f : T \to A) \times \mathsf{isEquiv}\ T\ A\ f$. If $f : \mathsf{Equiv}\ T\ A$ and $t : T$, we may write $f\ t$ for $f.1\ t$.

▶ **Lemma 7.** *If* $\Gamma \vdash f : \mathsf{Equiv}\ T\ A$, *we have an operation*

$$\frac{\Gamma, \varphi \vdash t : T \qquad \Gamma \vdash a : A \qquad \Gamma, \varphi \vdash p : \mathsf{Path}\ A\ a\ (f\ t)}{\Gamma \vdash \mathsf{equiv}\ f\ [\varphi \mapsto (t, p)]\ a : (x : T) \times \mathsf{Path}\ A\ a\ (f\ x)[\varphi \mapsto (t, p)]}$$

Conversely, if $\Gamma \vdash f : T \to A$ *and we have such an operation, then we can build a proof that* f *is an equivalence.*

Proof. We define $\mathsf{equiv}\ f\ [\varphi \mapsto (t, p)]\ a = \mathsf{contr}\ (f.2\ a)\ [\varphi \mapsto (t, p)]$ using the contr operation defined above. The second statement follows from Lemma 5. ◀

6 Glueing

In this section, we introduce the glueing operation. This operation expresses that to be "connected" is invariant by equivalence. From this operation, we can define a composition operation for universes, and prove the univalence axiom.

6.1 Syntax and inference rules for glueing

We introduce the *glueing* construction at type and term level by:

$$\begin{array}{llll}
t, u, A, B & ::= & \ldots & \\
& \mid & \mathsf{Glue}\ [\varphi \mapsto (T, f)]\ A & \text{Glue type} \\
& \mid & \mathsf{glue}\ [\varphi \mapsto t]\ u & \text{Glue term} \\
& \mid & \mathsf{unglue}\ [\varphi \mapsto (T, f)]\ u & \text{Unglue term}
\end{array}$$

We may write simply $\mathsf{unglue}\ b$ for $\mathsf{unglue}\ [\varphi \mapsto (T, f)]\ b$. The inference rules for these are presented in Figure 4.

It follows from these rules that if $\Gamma \vdash b : \mathsf{Glue}\ [\varphi \mapsto (T, f)]\ A$, then $\Gamma, \varphi \vdash b : T$.

$$\frac{\Gamma \vdash A \qquad \Gamma, \varphi \vdash T \qquad \Gamma, \varphi \vdash f : \mathsf{Equiv}\ T\ A}{\Gamma \vdash \mathsf{Glue}\ [\varphi \mapsto (T, f)]\ A} \qquad \frac{\Gamma \vdash b : \mathsf{Glue}\ [\varphi \mapsto (T, f)]\ A}{\Gamma \vdash \mathsf{unglue}\ b : A[\varphi \mapsto f\ b]}$$

$$\frac{\Gamma, \varphi \vdash f : \mathsf{Equiv}\ T\ A \qquad \Gamma, \varphi \vdash t : T \qquad \Gamma \vdash a : A[\varphi \mapsto f\ t]}{\Gamma \vdash \mathsf{glue}\ [\varphi \mapsto t]\ a : \mathsf{Glue}\ [\varphi \mapsto (T, f)]\ A}$$

$$\frac{\Gamma \vdash T \qquad \Gamma \vdash f : \mathsf{Equiv}\ T\ A}{\Gamma \vdash \mathsf{Glue}\ [1_{\mathbb{F}} \mapsto (T, f)]\ A = T} \qquad \frac{\Gamma \vdash t : T \qquad \Gamma \vdash a : A}{\Gamma \vdash \mathsf{glue}\ [1_{\mathbb{F}} \mapsto t]\ a = t : T}$$

$$\frac{\Gamma \vdash b : \mathsf{Glue}\ [\varphi \mapsto (T, f)]\ A}{\Gamma \vdash b = \mathsf{glue}\ [\varphi \mapsto b]\ (\mathsf{unglue}\ b) : \mathsf{Glue}\ [\varphi \mapsto (T, f)]\ A}$$

$$\frac{\Gamma, \varphi \vdash t : T \qquad \Gamma \vdash a : A[\varphi \mapsto f\ t]}{\Gamma \vdash \mathsf{unglue}\ (\mathsf{glue}\ [\varphi \mapsto t]\ a) = a : A}$$

Figure 4 Inference rules for glueing

In the case $\varphi = (i = 0) \vee (i = 1)$ the glueing operation can be illustrated as the dashed line in:

$$\begin{tikzcd}
T_0 \arrow[r, dashed] \arrow[d, "f(i0)"', "\wr"] & T_1 \arrow[d, "f(i1)", "\wr"'] \\
A(i0) \arrow[r, "A"'] & A(i1)
\end{tikzcd}$$

This illustrates why the operation is called glue: it *glues* together along a partial equivalence the partial type T and the total type A to a total type that extends T.

Using glueing we can construct a path from an equivalence:

▶ **Example 8.** Given $\Gamma \vdash f : \mathsf{Equiv}\ A\ B$ we define

$$\Gamma, i : \mathbb{I} \vdash E = \mathsf{Glue}\ [(i = 0) \mapsto (A, f), (i = 1) \mapsto (B, \mathsf{id}_B)]\ B$$

so that $E(i0) = A$ and $E(i1) = B$, where $\mathsf{id}_B : \mathsf{Equiv}\ B\ B$ is defined as

$$\mathsf{id}_B = (\lambda x : B.\ x, \lambda x : B.\ ((x, 1_x), \lambda u : (y : B) \times \mathsf{Path}\ B\ x\ y.\ \langle i \rangle\ (u.2\ i, \langle j \rangle\ u.2\ (i \wedge j))))$$

We can then define a function $(A\ B : \mathsf{U}) \to \mathsf{Equiv}\ A\ B \to \mathsf{Path}\ \mathsf{U}\ A\ B$ by:

$$\lambda A\ B : \mathsf{U}.\ \lambda f : \mathsf{Equiv}\ A\ B.\ \langle i \rangle\ \mathsf{Glue}\ [(i = 0) \mapsto (A, f), (i = 1) \mapsto (B, \mathsf{id}_B)]\ B$$

6.2 Composition for glueing

We assume $\Gamma, i : \mathbb{I} \vdash B = \mathsf{Glue}\ [\varphi \mapsto (T, f)]\ A$, and define the composition in B. In order to do so, assume

$$\Gamma, \psi, i : \mathbb{I} \vdash b : B \qquad\qquad \Gamma \vdash b_0 : B(i0)[\psi \mapsto b(i0)]$$

and define:

$$\Gamma, \psi, i : \mathbb{I} \vdash a = \mathsf{unglue}\ b : A[\varphi \mapsto f\ b]$$
$$\Gamma \vdash a_0 = \mathsf{unglue}\ b_0 : A(i0)[\varphi(i0) \mapsto f(i0)\ b_0, \psi \mapsto a(i0)]$$

The following provides the algorithm for composition $b_1 = \mathsf{comp}^i\ B\ [\psi \mapsto b]\ b_0$ of type $B(i1)[\psi \mapsto b(i1)]$.

$$\begin{array}{lcll}
\delta & = & \forall i.\varphi & \Gamma \\
a'_1 & = & \mathsf{comp}^i\ A\ [\psi \mapsto a]\ a_0 & \Gamma \\
t'_1 & = & \mathsf{comp}^i\ T\ [\psi \mapsto b]\ b_0 & \Gamma, \delta \\
\omega & = & \mathsf{pres}^i\ f\ [\psi \mapsto b]\ b_0 & \Gamma, \delta \\
(t_1, \alpha) & = & \mathsf{equiv}\ f(i1)\ [\delta \mapsto (t'_1, \omega), \psi \mapsto (b(i1), \langle j \rangle a'_1)]\ a'_1 & \Gamma, \varphi(i1) \\
a_1 & = & \mathsf{comp}^j\ A(i1)\ [\varphi(i1) \mapsto \alpha\ j, \psi \mapsto a(i1)]\ a'_1 & \Gamma \\
b_1 & = & \mathsf{glue}\ [\varphi(i1) \mapsto t_1]\ a_1 & \Gamma
\end{array}$$

On the extent δ we have $B = T$ and we can check $\Gamma, \delta \vdash b_1 = \mathsf{comp}^i\ T\ [\psi \mapsto b]\ b_0 : T(i1)$.

In the next section we will use the glue operation to define the composition for the universe and to prove the univalence axiom.

7 Universe and the univalence axiom

As in [18], we now introduce a universe U à la Russell by reflecting all typing rules and

$$\frac{\Gamma \vdash}{\Gamma \vdash \mathsf{U}} \qquad\qquad \frac{\Gamma \vdash A : \mathsf{U}}{\Gamma \vdash A}$$

In particular, we have $\Gamma \vdash \mathsf{Glue}\ [\varphi \mapsto (T, f)]\ A : \mathsf{U}$ whenever $\Gamma \vdash A : \mathsf{U}$, $\Gamma, \varphi \vdash T : \mathsf{U}$, and $\Gamma, \varphi \vdash f : \mathsf{Equiv}\ T\ A$.

7.1 Composition for the universe

In order to describe the composition operation for the universe we first have to explain how to construct an equivalence from a line in the universe. Given $\Gamma \vdash A$, $\Gamma \vdash B$, and $\Gamma, i : \mathbb{I} \vdash E$, such that $E(i0) = A$ and $E(i1) = B$, we will construct $\mathsf{equiv}^i\ E : \mathsf{Equiv}\ A\ B$. In order to do this we first define

$$\begin{array}{rcl}
\Gamma \vdash f & = & \lambda x : A.\ \mathsf{transp}^i\ \ E\ x : A \to B \\
\Gamma \vdash g & = & \lambda y : B.\ (\mathsf{transp}^i\ \ E(i/1-i)\ y)(i/1-i) : B \to A \\
\Gamma, i : \mathbb{I} \vdash u & = & \lambda x : A.\ \mathsf{fill}^i\ E\ []\ x : A \to E \\
\Gamma, i : \mathbb{I} \vdash v & = & \lambda y : B.\ (\mathsf{fill}^i\ E(i/1-i)\ []\ y)(i/1-i) : B \to E
\end{array}$$

such that:

$$u(i0) = \lambda x : A.x \qquad u(i1) = f \qquad v(i0) = g \qquad v(i1) = \lambda y : B.y$$

We will now prove that f is an equivalence. Given $y : B$ we see that $(x : A) \times \mathsf{Path}\ B\ y\ (f\ x)$ is inhabited as it contains the element $(g\ y, \gamma)$ where

$$\gamma = \langle j \rangle\ \mathsf{comp}^i\ E\ [(j = 0) \mapsto v\ y, (j = 1) \mapsto u\ (g\ y)]\ (g\ y)$$

We then show that two elements (x_0, β_0) and (x_1, β_1) in $(x : A) \times \mathsf{Path}\ B\ y\ (f\ x)$ are path-connected. This is obtained by the definitions

$$
\begin{array}{lcl}
\omega_0 & = & \mathsf{comp}^i\ E(i/1-i)\ [(j=0) \mapsto v\ y, (j=1) \mapsto u\ x_0]\ (\beta_0\ j) \\
\omega_1 & = & \mathsf{comp}^i\ E(i/1-i)\ [(j=0) \mapsto v\ y, (j=1) \mapsto u\ x_1]\ (\beta_1\ j) \\
\theta_0 & = & \mathsf{fill}^i\ E(i/1-i)\ [(j=0) \mapsto v\ y, (j=1) \mapsto u\ x_0]\ (\beta_0\ j) \\
\theta_1 & = & \mathsf{fill}^i\ E(i/1-i)\ [(j=0) \mapsto v\ y, (j=1) \mapsto u\ x_1]\ (\beta_1\ j) \\
\omega & = & \mathsf{comp}^j\ A\ [(k=0) \mapsto \omega_0, (k=1) \mapsto \omega_1]\ (g\ y) \\
\theta & = & \mathsf{fill}^j\ A\ [(k=0) \mapsto \omega_0, (k=1) \mapsto \omega_1]\ (g\ y)
\end{array}
$$

so that we have $\Gamma, j : \mathbb{I}, i : \mathbb{I} \vdash \theta_0 : E$ and $\Gamma, j : \mathbb{I}, i : \mathbb{I} \vdash \theta_1 : E$ and $\Gamma, j : \mathbb{I}, k : \mathbb{I} \vdash \theta : A$. If we define

$$\delta = \mathsf{comp}^i\ E\ [(j=0) \mapsto v\ y, (j=1) \mapsto u\ \omega, (k=0) \mapsto \theta_0, (k=1) \mapsto \theta_1]\ \theta$$

we then have

$$\langle k \rangle\ (\omega, \langle j \rangle \delta) : \mathsf{Path}\ ((x : A) \times \mathsf{Path}\ B\ y\ (f\ x))\ (x_0, \beta_0)\ (x_1, \beta_1)$$

as desired. We have hence shown that f is an equivalence, so we have constructed $\mathsf{equiv}^i\ E : \mathsf{Equiv}\ A\ B$.

Using this we can now define the composition for the universe:

$$\Gamma \vdash \mathsf{comp}^i\ \mathsf{U}\ [\varphi \mapsto E]\ A_0 = \mathsf{Glue}\ [\varphi \mapsto (E(i1), \mathsf{equiv}^i\ E(i/1-i))]\ A_0 : \mathsf{U}[\varphi \mapsto E(i1)]$$

7.2 The univalence axiom

Given $B = \mathsf{Glue}\ [\varphi \mapsto (T, f)]\ A$ the map $\mathsf{unglue} : B \to A$ extends f, in the sense that $\Gamma, \varphi \vdash \mathsf{unglue}\ b = f\ b : A$ if $\Gamma \vdash b : B$.

▶ **Theorem 9.** *The map* $\mathsf{unglue} : B \to A$ *is an equivalence.*

Proof. By Lemma 7 it suffices to construct

$$\tilde{b} : B[\psi \mapsto b] \qquad\qquad \tilde{\alpha} : \mathsf{Path}\ A\ u\ (\mathsf{unglue}\ \tilde{b})[\psi \mapsto \alpha]$$

given $\Gamma, \psi \vdash b : B$ and $\Gamma \vdash u : A$ and $\Gamma, \psi \vdash \alpha : \mathsf{Path}\ A\ u\ (\mathsf{unglue}\ b)$.

Since $\Gamma, \varphi \vdash f : T \to A$ is an equivalence and

$$\Gamma, \varphi, \psi \vdash b : T \qquad\qquad \Gamma, \varphi, \psi \vdash \alpha : \mathsf{Path}\ A\ u\ (f\ b)$$

we get, using Lemma 7

$$\Gamma, \varphi \vdash t : T[\psi \mapsto b] \qquad\qquad \Gamma, \varphi \vdash \beta : \mathsf{Path}\ A\ u\ (f\ t)\ [\psi \mapsto \alpha]$$

We then define $\tilde{a} = \mathsf{comp}^i\ A\ [\varphi \mapsto \beta\ i, \psi \mapsto \alpha\ i]\ u$, and using this we conclude by letting $\tilde{b} = \mathsf{glue}([\varphi \mapsto t], \tilde{a})$ and $\tilde{\alpha} = \mathsf{fill}^i\ A\ [\varphi \mapsto \beta\ i, \psi \mapsto \alpha\ i]\ u$. ◀

▶ **Corollary 10.** *For any type* $A : \mathsf{U}$ *the type* $C = (X : \mathsf{U}) \times \mathsf{Equiv}\ X\ A$ *is contractible.*[4]

[4] This formulation of the univalence axiom can be found in the message of Martín Escardó in: `https://groups.google.com/forum/#!msg/homotopytypetheory/HfCB_b-PNEU/Ibb48LvUMeUJ`
This is also used in the (classical) proofs of the univalence axiom, see Theorem 3.4.1 of [15] and Proposition 2.18 of [8], where an operation similar to the glueing operation appears implicitly.

Proof. It is enough by Lemma 5 to show that any partial element $\varphi \vdash (T, f) : C$ is path equal to the restriction of a total element. The map unglue extends f and is an equivalence by the previous theorem. Since any two elements of the type $\mathsf{isEquiv}\ X\ A\ f.1$ are path equal, this shows that any partial element of type C is path equal to the restriction of a total element. We can then conclude by Theorem 9. ◀

▶ **Corollary 11** (Univalence axiom)**.** *For any term*

$$t : (A\ B : \mathsf{U}) \to \mathsf{Path}\ \mathsf{U}\ A\ B \to \mathsf{Equiv}\ A\ B$$

the map $t\ A\ B : \mathsf{Path}\ \mathsf{U}\ A\ B \to \mathsf{Equiv}\ A\ B$ *is an equivalence.*

Proof. Both $(X : \mathsf{U}) \times \mathsf{Path}\ \mathsf{U}\ A\ X$ and $(X : \mathsf{U}) \times \mathsf{Equiv}\ A\ X$ are contractible. Hence the result follows from Theorem 4.7.7 in [24]. ◀

An alternative proof of univalence can be found in Appendix B.

8 Semantics

In this section we will explain the semantics of the type theory under consideration in cubical sets. We will first review how cubical sets, as a presheaf category, yield a model of basic type theory, and then explain the additional so-called composition structure we have to require to interpret the full cubical type theory.

8.1 The category of cubes and cubical sets

Consider the monad dM on the category of sets associating to each set the free de Morgan algebra on that set. The *category of cubes* $\mathcal{C}$ is the small category whose objects are finite subsets $I, J, K, \dots$ of a fixed, discrete, and countably infinite set, called *names*, and a morphism $\mathrm{Hom}(J, I)$ is a map $I \to \mathsf{dM}(J)$. Identities and compositions are inherited from the Kleisli category of dM, i.e., the identity on I is given by the unit $I \to \mathsf{dM}(I)$, and composition $fg \in \mathrm{Hom}(K, I)$ of $g \in \mathrm{Hom}(K, J)$ and $f \in \mathrm{Hom}(J, I)$ is given by $\mu_K \circ \mathsf{dM}(g) \circ f$ where $\mu_K : \mathsf{dM}(\mathsf{dM}(K)) \to \mathsf{dM}(K)$ denotes multiplication of dM. We will use f, g, h for morphisms in $\mathcal{C}$ and simply write $f : J \to I$ for $f \in \mathrm{Hom}(J, I)$. We will often write unions with commas and omit curly braces around finite sets of names, e.g., writing I, i, j for $I \cup \{i, j\}$ and $I - i$ for $I - \{i\}$ etc.

If i is in I, we have maps (ib) in $\mathrm{Hom}(I - i, I)$ sending i to b, for $b = 0_{\mathbb{I}}$ or $1_{\mathbb{I}}$. A *face map* is a composition of such maps. A *strict map* $\mathrm{Hom}(J, I)$ is a map $I \to \mathsf{dM}(J)$ which never takes the value $0_{\mathbb{I}}$ or $1_{\mathbb{I}}$. Any f can be uniquely written as a composition $f = gh$ where g is a face map and h is strict.

▶ **Definition 12.** A *cubical set* is a presheaf on $\mathcal{C}$.

Thus, a cubical set Γ is given by sets $\Gamma(I)$ for each $I \in \mathcal{C}$ and maps (called restrictions) $\Gamma(f) : \Gamma(I) \to \Gamma(J)$ for each $f : J \to I$. If we write $\Gamma(f)(\rho) = \rho f$ for $\rho \in \Gamma(I)$ (leaving the Γ implicit), these maps should satisfy $\rho\,\mathrm{id}_I = \rho$ and $(\rho f)g = \rho(fg)$ for $f : J \to I$ and $g : K \to J$.

Let us discuss some important examples of cubical sets. Using the canonical de Morgan algebra structure of the unit interval, $[0, 1]$, we can define a functor

$$\mathcal{C} \to \mathbf{Top}, \quad I \mapsto [0, 1]^I. \tag{1}$$

If u is in $[0,1]^I$ we can think of u as an environment giving values in $[0,1]$ to each $i \in I$, so that iu is in $[0,1]$ if $i \in I$. Since $[0,1]$ is a de Morgan algebra, this extends uniquely to ru for $r \in \mathsf{dM}(I)$. So any $f\colon J \to I$ in $\mathcal{C}$ induces $f\colon [0,1]^J \to [0,1]^I$ by $i(fu) = (if)u$.

To any topological space X we can associate its *singular cubical set* $\mathrm{S}(X)$ by taking $\mathrm{S}(X)(I)$ to be the set of continuous functions $[0,1]^I \to X$.

For a finite set of names I we get the formal cube $\mathbf{y}\,I$ where $\mathbf{y}\colon \mathcal{C} \to [\mathcal{C}^{\mathrm{op}}, \mathbf{Set}]$ denotes the Yoneda embedding. Note that since **Top** is cocomplete the functor in (1) extends to a cocontinuous functor assigning to each cubical set its *geometric realization* as a topological space, in such a way that $\mathbf{y}\,I$ has $[0,1]^I$ as its geometric realization.

The formal interval $\mathbb{I}$ induces a cubical set given by $\mathbb{I}(I) = \mathsf{dM}(I)$. The face lattice $\mathbb{F}$ induces a cubical set by taking as $\mathbb{F}(I)$ to be those $\varphi \in \mathbb{F}$ which only use symbols in I. The restrictions along $f\colon J \to I$ are in both cases simply *substituting* the symbols $i \in I$ by $f(i) \in \mathsf{dM}(J)$.

As any presheaf category, cubical sets have a subobject classifier Ω where $\Omega(I)$ is the set of sieves on I (i.e., subfunctors of $\mathbf{y}\,I$). Consider the natural transformation $(\cdot = 1)\colon \mathbb{I} \to \Omega$ where for $r \in \mathbb{I}(I)$, $(r = 1)$ is the sieve on I of all $f\colon J \to I$ such that $rf = 1_{\mathbb{I}}$. The image of $(\cdot = 1)$ is $\mathbb{F} \to \Omega$, assigning to each φ the sieve of all f with $\varphi f = 1_{\mathbb{F}}$.

8.2 Presheaf semantics

The category of cubical sets (with morphisms being natural transformations) induce—as does any presheaf category—a category with families (CwF) [9] where the category of contexts and substitutions is the category of cubical sets. We will review the basic constructions but omit verification of the required equations (see, e.g., [12, 13, 6] for more details).

Basic presheaf semantics

As already mentioned the category of (semantic) contexts and substitutions is given by cubical sets and their maps. In this section we will use Γ, Δ to denote cubical sets and (semantic) substitutions by $\sigma\colon \Delta \to \Gamma$, overloading previous use of the corresponding meta-variables to emphasize their intended role.

Given a cubical set Γ, the types A in context Γ, written $A \in \mathrm{Ty}(\Gamma)$, are given by sets $A\rho$ for each $I \in \mathcal{C}$ and $\rho \in \Gamma(I)$ together with restriction maps $A\rho \to A(\rho f)$, $u \mapsto uf$ for $f\colon J \to I$ satisfying $u\,\mathrm{id}_I = u$ and $(uf)g = u(fg) \in A(\rho fg)$ if $g\colon K \to J$. Equivalently, $A \in \mathrm{Ty}(\Gamma)$ are the presheaves on the category of elements of Γ. For a type $A \in \mathrm{Ty}(\Gamma)$ its terms $a \in \mathrm{Ter}(\Gamma; A)$ are given by families of elements $a\rho \in A\rho$ for each $I \in \mathcal{C}$ and $\rho \in \Gamma(I)$ such that $(a\rho)f = a(\rho f)$ for $f\colon J \to I$. Note that our notation leaves a lot implicit; e.g., we should have written $A(I, \rho)$ for $A\rho$; $A(I, \rho, f)$ for the restriction map $A\rho \to A(\rho f)$; and $a(I, \rho)$ for $a\rho$.

For $A \in \mathrm{Ty}(\Gamma)$ and $\sigma\colon \Delta \to \Gamma$ we define $A\sigma \in \mathrm{Ty}(\Delta)$ by $(A\sigma)\rho = A(\sigma\rho)$ and the induced restrictions. If we also have $a \in \mathrm{Ter}(\Gamma; A)$, we define $a\sigma \in \mathrm{Ter}(\Delta; A\sigma)$ by $(a\sigma)\rho = a(\rho\sigma)$. For a type $A \in \mathrm{Ty}(\Gamma)$ we define the cubical set $\Gamma.A$ by $(\Gamma.A)(I)$ being the set of all (ρ, u) with $\rho \in \Gamma(I)$ and $u \in A\rho$; restrictions are given by $(\rho, u)f = (\rho f, uf)$. The first projection yields a map $\mathsf{p}\colon \Gamma.A \to \Gamma$ and the second projection a term $\mathsf{q} \in \mathrm{Ter}(\Gamma; A\mathsf{p})$. Given $\sigma\colon \Delta \to \Gamma$, $A \in \mathrm{Ty}(\Gamma)$, and $a \in \mathrm{Ter}(\Delta; A\sigma)$ we define $(\sigma, a)\colon \Delta \to \Gamma.A$ by $(\sigma, a)\rho = (\sigma\rho, a\rho)$. For $u \in \mathrm{Ter}(\Gamma; A)$ we define $[u] = (\mathrm{id}_\Gamma, u)\colon \Gamma \to \Gamma.A$.

The basic type formers are interpreted as follows. For $A \in \mathrm{Ty}(\Gamma)$ and $B \in \mathrm{Ty}(\Gamma.A)$ define $\Sigma_\Gamma(A, B) \in \mathrm{Ty}(\Gamma)$ by letting $\Sigma_\Gamma(A, B)\rho$ contain all pairs (u, v) where $u \in A\rho$ and $v \in B(\rho, v)$; restrictions are defined as $(u, v)f = (uf, vf)$. Given $w \in \mathrm{Ter}(\Gamma; \Sigma(A, B))$ we get $w.1 \in \mathrm{Ter}(\Gamma; A)$ and $w.2 \in \mathrm{Ter}(\Gamma; B[w.1])$ by $(w.1)\rho = \mathsf{p}(w\rho)$ and $(w.2)\rho = \mathsf{q}(w\rho)$ where $\mathsf{p}(u, v) = u$ and $\mathsf{q}(u, v) = v$ are the set-theoretic projections.

Given $A \in \mathrm{Ty}(\Gamma)$ and $B \in \mathrm{Ty}(\Gamma.A)$ the dependent function space $\Pi_\Gamma(A, B) \in \mathrm{Ty}(\Gamma)$ is defined by letting $\Pi_\Gamma(A, B)\rho$ for $\rho \in \Gamma(I)$ contain all families $w = (w_f \mid J \in \mathcal{C}, f \colon J \to I)$ where

$$w_f \in \prod_{u \in A(\rho f)} B(\rho f, u) \quad \text{such that} \quad (w_f\, u)g = w_{fg}(ug) \quad \text{for } u \in A(\rho f),\ g \colon K \to J.$$

The restriction by $f \colon J \to I$ of such a w is defined by $(wf)_g = w_{fg}$. Given $v \in \mathrm{Ter}(\Gamma.A; B)$ we have $\lambda_{\Gamma;A} v \in \mathrm{Ter}(\Gamma; \Pi(A, B))$ given by $((\lambda v)\rho)_f\, u = v(\rho f, u)$. Application $\mathtt{app}(w, u) \in \mathrm{Ter}(\Gamma; B[u])$ of $w \in \mathrm{Ter}(\Gamma; \Pi(A, B))$ to $u \in \mathrm{Ter}(\Gamma; A)$ is defined by

$$\mathtt{app}(w, u)\rho = (w\rho)_{\mathrm{id}_I}(u\rho) \in (B[u])\rho. \tag{2}$$

Basic data types like the natural numbers can be interpreted as discrete presheaves, i.e., $\mathtt{N} \in \mathrm{Ty}(\Gamma)$ is given by $\mathtt{N}\rho = \mathbb{N}$; the constants are interpreted by the lifts of the corresponding set-theoretic operations on $\mathbb{N}$. This concludes the outline of the basic CwF structure on cubical sets.

▶ Remark. Following Aczel [1] we will make use of that our semantic entities are actual sets in the ambient set theory. This will allow us to interpret syntax in Section 8.3 with fewer type annotations than are usually needed for general categorical semantics of type theory (see [22]). E.g., the definition of application $\mathtt{app}(w, u)\rho$ as defined in (2) is independent of Γ, A and B, since set-theoretic application is a (class) operation on all sets. Likewise, we don't need annotations for first and second projections. But note that we will need the type A for λ-abstraction for $(\lambda_{\Gamma;A} v)\rho$ to be a set by the replacement axiom.

Semantic path types

Note that we can consider any cubical set X as $X' \in \mathrm{Ty}(\Gamma)$ by setting $X'\rho = X(I)$ for $\rho \in \Gamma(I)$. We will usually simply write X for X'. In particular, for a cubical set Γ we can form the cubical set $\Gamma.\mathbb{I}$.

For $A \in \mathrm{Ty}(\Gamma)$ and $u, v \in \mathrm{Ter}(\Gamma; A)$ the semantic path type $\mathtt{Path}^\Gamma_A(u, v) \in \mathrm{Ty}(\Gamma)$ is given by: for $\rho \in \Gamma(I)$, $\mathtt{Path}_A(u, v)\rho$ consists of equivalence classes $\langle i\rangle\ w$ where $i \notin I$, $w \in A(\rho \mathrm{s}_i)$ such that $w(i0) = u\rho$ and $w(i1) = v\rho$; two such elements $\langle i\rangle\ w$ and $\langle j\rangle\ w'$ are equal iff $w(i/j) = w'$. Here $\mathrm{s}_i \colon I, i \to I$ is induced by the inclusion $I \subseteq I, i$ and (i/j) setting i to j. We define $(\langle i\rangle\ w)f = \langle j\rangle\ w(f, i = j)$ for $f \colon J \to I$ and $j \notin J$. For $\varphi \in \mathbb{F}(I)$ we set $(\langle i\rangle\ w)\varphi = w(i/\varphi)$. Both operations, name abstraction and application, lift to terms, i.e., if $w \in \mathrm{Ter}(\Gamma.\mathbb{I}; A)$, then $\langle\,\rangle w \in \mathrm{Ter}(\Gamma; \mathtt{Path}_A(w[0], w[1]))$ given by $(\langle\,\rangle w)\rho = \langle i\rangle\ w(\rho \mathrm{s}_i)$ for a fresh i; also if $u \in \mathrm{Ter}(\Gamma; \mathtt{Path}_A(a, b))$ and $\varphi \in \mathrm{Ter}(\Gamma; \mathbb{F})$, then $u\varphi \in \mathrm{Ter}(\Gamma; A)$ defined as $(u\varphi)\rho = (u\rho)(\varphi\rho)$.

Composition structure

For $\varphi \in \mathrm{Ter}(\Gamma; \mathbb{F})$ we define the cubical set Γ, φ by taking $\rho \in (\Gamma, \varphi)(I)$ iff $\rho \in \Gamma(I)$ and $\varphi\rho = 1_\mathbb{F} \in \mathbb{F}$; the restrictions are those induced by Γ. In particular, we have $\Gamma, 1 = \Gamma$ and $\Gamma, 0$ is the empty cubical set. (Here, $0 \in \mathrm{Ter}(\Gamma; \mathbb{F})$ is $0\rho = 0_\mathbb{F}$ and similarly for $1_\mathbb{F}$.) Any $\sigma \colon \Delta \to \Gamma$ gives rise to a morphism $\Delta, \varphi\sigma \to \Gamma, \varphi$ which we also will denote by σ.

If $A \in \mathrm{Ty}(\Gamma)$ and $\varphi \in \mathrm{Ter}(\Gamma; \mathbb{F})$, we define a *partial element of $A \in \mathrm{Ty}(\Gamma)$ of extent φ* to be an element of $\mathrm{Ter}(\Gamma, \varphi; A\iota_\varphi)$ where $\iota_\varphi \colon \Gamma, \varphi \hookrightarrow \Gamma$ is the inclusion. So, such a partial element u is given by a family of elements $u\rho \in A\rho$ for each $\rho \in \Gamma(I)$ such that $\varphi\rho = 1$, satisfying $(u\rho)f = u(\rho f)$ whenever $f\colon J \to I$. Each $u \in \mathrm{Ter}(\Gamma; A)$ gives rise to the partial element $u\iota \in \mathrm{Ter}(\Gamma, \varphi; A\iota)$; a partial element is *connected* if it is induced by such an element of $\mathrm{Ter}(\Gamma; A)$.

For the next definition note that if $A \in \mathrm{Ty}(\Gamma)$, then $\rho \in \Gamma(I)$ corresponds to $\rho\colon\ \mathbf{y}\, I \to \Gamma$ and thus $A\rho \in \mathrm{Ty}(\mathbf{y}\, I)$; also, any $\varphi \in \mathbb{F}(I)$ corresponds to $\varphi \in \mathrm{Ter}(\mathbf{y}\, I; \mathbb{F})$.

▶ **Definition 13.** A *composition structure* for $A \in \mathrm{Ty}(\Gamma)$ is given by the following operations. For each I, $i \notin I$, $\rho \in \Gamma(I, i)$, $\varphi \in \mathbb{F}(I)$, u a partial element of $A\rho$ of extent φ, and $a_0 \in A\rho(i0)$ with $a_0 f = u_f$ for all f with $\varphi f = 1_{\mathbb{F}}$ (i.e., $a_0\iota_\varphi = u$ if a_0 is considered as element of $\mathrm{Ter}(\mathbf{y}\, I; A\rho)$), we require

$$\mathtt{comp}(I, i, \rho, \varphi, u, a_0) \in A\rho(i1)$$

such that for any $f\colon I \to J$ and $j \notin J$,

$$(\mathtt{comp}(I, i, \rho, \varphi, u, a_0))f = \mathtt{comp}(J, j, \rho(f, i = j), \varphi f, u(f, i = j), a_0 f),$$

and $\mathtt{comp}(I, i, \rho, 1_{\mathbb{F}}, u, a_0) = u_{\mathrm{id}_I}$.

A type $A \in \mathrm{Ty}(\Gamma)$ together with a composition structure $\mathtt{comp}$ on A is called a *fibrant type*, written $(A, \mathtt{comp}) \in \mathrm{FTy}(\Gamma)$. We will usually simply write $A \in \mathrm{FTy}(\Gamma)$ and $\mathtt{comp}_A$ for its composition structure. But observe that $A \in \mathrm{Ty}(\Gamma)$ can have different composition structures. Call a cubical set Γ *fibrant* if it is a fibrant type when Γ considered as type $\Gamma \in \mathrm{Ty}(\top)$ is fibrant where $\top$ is a terminal cubical set. A prime example of a fibrant cubical set is the singular cubical set of a topological space (see Appendix C).

▶ **Theorem 14.** *The CwF on cubical sets supporting dependent products, dependent sums, and natural numbers described above can be extended to fibrant types.*

Proof. For example, if $A \in \mathrm{FTy}(\Gamma)$ and $\sigma\colon \Delta \to \Gamma$, we set

$$\mathtt{comp}_{A\sigma}(I, i, \rho, \varphi, u, a_0) = \mathtt{comp}_A(I, i, \sigma\rho, \varphi, u, a_0)$$

as the composition structure on $A\sigma$ in $\mathrm{FTy}(\Delta)$. Type formers are treated analogously to their syntactic counterpart given in Section 4. Note that one also has to check that all equations between types are also preserved by their associated composition structures. ◀

Note that we can also, like in the syntax, define a composition structure on $\mathtt{Path}_A(u, v)$ given that A has one.

Semantic glueing

Next we will give a semantic counterpart to the Glue construction. To define the semantic glueing as an element of $\mathrm{Ty}(\Gamma)$ it is not necessary that the given types have composition structures or that the functions are equivalences; this is only needed later to give the composition structure. Assume $\varphi \in \mathrm{Ter}(\Gamma; \mathbb{F})$, $T \in \mathrm{Ty}(\Gamma, \varphi)$, $A \in \mathrm{Ty}(\Gamma)$, and $w \in \mathrm{Ter}(\Gamma, \varphi; T \to A\iota)$ (where $A \to B$ is $\Pi(A, B\mathtt{p})$).

▶ **Definition 15.** The *semantic glueing* $\mathtt{Glue}_\Gamma(\varphi, T, A, w) \in \mathrm{Ty}(\Gamma)$ is defined as follows. For $\rho \in \Gamma(I)$, we let $u \in \mathtt{Glue}(\varphi, T, A, w)\rho$ iff either

- $u \in T\rho$ and $\varphi\rho = 1_{\mathbb{F}}$; or

- $u = \mathtt{glue}(\varphi\rho, t, a)$ and $\varphi\rho \neq 1_{\mathbb{F}}$, where $t \in \mathrm{Ter}(\mathbf{y}\,I, \varphi\rho; T\rho)$ and $a \in \mathrm{Ter}(\mathbf{y}\,I; A\rho)$ such that $\mathtt{app}(w\rho, t) = a\iota \in \mathrm{Ter}(\mathbf{y}\,I, \varphi\rho; A\rho\iota)$.

For $f\colon J \to I$ we define the restriction uf of $u \in \mathtt{Glue}(\varphi, T, A, w)$ to be given by the restriction of $T\rho$ in the first case; in the second case, i.e., if $\varphi\rho \neq 1_{\mathbb{F}}$, we let $uf = \mathtt{glue}(\varphi\rho, t, a)f = t_f \in T\rho f$ in case $\varphi\rho f = 1_{\mathbb{F}}$, and otherwise $uf = \mathtt{glue}(\varphi\rho f, tf, af)$.

Here glue was defined as a constructor; we extend glue to any $t \in \mathrm{Ter}(\mathbf{y}\,I; T\rho)$, $a \in \mathrm{Ter}(\mathbf{y}\,I; A\rho)$ such that $\mathtt{app}(w\rho, t) = a$ (so if $\varphi\rho = 1_{\mathbb{F}}$) by $\mathtt{glue}(1_{\mathbb{F}}, t, a) = t_{\mathrm{id}_I}$. This way any element of $\mathtt{Glue}(\varphi, T, A, w)\rho$ is of the form $\mathtt{glue}(\varphi\rho, t, a)$ for suitable t and a, and restriction is given by $(\mathtt{glue}(\varphi\rho, t, a))f = \mathtt{glue}(\varphi\rho f, tf, af)$. Note that we get

$$\mathtt{Glue}_\Gamma(1_{\mathbb{F}}, T, A, w) = T \text{ and } (\mathtt{Glue}_\Gamma(\varphi, T, A, w))\sigma = \mathtt{Glue}_\Delta(\varphi\sigma, T\sigma, A\sigma, w\sigma) \text{ for } \sigma\colon \Delta \to \Gamma. \tag{3}$$

We define $\mathtt{unglue}(\varphi, T, w) \in \mathrm{Ter}(\Gamma.\,\mathtt{Glue}(\varphi, T, A, w); A\mathtt{p})$ by

$$\begin{aligned} \mathtt{unglue}(\varphi, T, w)(\rho, t) &= \mathtt{app}(w\rho, t)_{\mathrm{id}_I} \in A\rho && \text{whenever } \varphi\rho = 1_{\mathbb{F}}, \text{ and} \\ \mathtt{unglue}(\varphi, T, w)(\rho, \mathtt{glue}(\varphi, t, a)) &= a && \text{otherwise,} \end{aligned}$$

where $\rho \in \Gamma(I)$.

▶ **Definition 16.** For $A, B \in \mathrm{Ty}(\Gamma)$ and $w \in \mathrm{Ter}(\Gamma; A \to B)$ an *equivalence structure* for w is given by the following operations such that for each
- $\rho \in \Gamma(I)$,
- $\varphi \in \mathbb{F}(I)$,
- $b \in B\rho$, and
- partial elements a of $A\rho$ and ω of $\mathtt{Path}_B(\mathtt{app}(w\rho, a), b\iota)\rho$ with extent φ,

we are given

$$\mathtt{e}_0(\rho, \varphi, b, a, \omega) \in A\rho, \text{ and a path } \mathtt{e}_1(\rho, \varphi, b, a, \omega) \text{ between } \mathtt{app}(w\rho, \mathtt{e}_0(\rho, \varphi, b, a, \omega)) \text{ and } b$$

such that $\mathtt{e}_0(\rho, \varphi, b, a, \omega)\iota = a$, $\mathtt{e}_1(\rho, \varphi, b, a, \omega)\iota = \omega$ (where $\iota\colon \mathbf{y}\,I, \varphi \to \mathbf{y}\,I$) and for any $f\colon J \to I$ and $b = 0, 1$:

$$(\mathtt{e}_b(\rho, \varphi, b, a, \omega))f = \mathtt{e}_b(\rho f, \varphi f, bf, af, \omega f).$$

Following the argument in the syntax we can use the equivalence structure to explain a composition for Glue.

▶ **Theorem 17.** *If $A \in \mathrm{FTy}(\Gamma)$, $T \in \mathrm{FTy}(\Gamma, \varphi)$, and we have an equivalence structure for w, then we have a composition structure for $\mathtt{Glue}(\varphi, T, A, w)$ such that the equations* (3) *also hold for the respective composition structures.*

Semantic universes

Assuming a Grothendieck universe of small sets in our ambient set theory, we can define $A \in \mathrm{Ty}_0(\Gamma)$ iff all $A\rho$ are small for $\rho \in \Gamma(I)$; and $A \in \mathrm{FTy}_0(\Gamma)$ iff $A \in \mathrm{Ty}_0(\Gamma)$ when forgetting the composition structure of A.

▶ **Definition 18.** The semantic universe $\mathtt{U}$ is the cubical set defined by $\mathtt{U}(I) = \mathrm{FTy}_0(\mathbf{y}\,I)$; restriction along $f\colon J \to I$ is simply substitution along $\mathbf{y}\,f$.

We can consider $\mathtt{U}$ as an element of $\mathrm{Ty}(\Gamma)$. As such we can, as in the syntactic counterpart, define a composition structure on $\mathtt{U}$ using semantic glueing, so that $\mathtt{U} \in \mathrm{FTy}(\Gamma)$. Note that semantic glueing preserves smallness.

For $T \in \mathrm{Ter}(\Gamma; \mathtt{U})$ we can define decoding $\mathsf{El}\, T \in \mathrm{FTy}_0(\Gamma)$ by $(\mathsf{El}\, T)\rho = (T\rho)\, \mathrm{id}_I$ and likewise for the composition structure. For $A \in \mathrm{FTy}_0(\Gamma)$ we get its code $\ulcorner A \urcorner \in \mathrm{Ter}(\Gamma; \mathtt{U})$ by setting $\ulcorner A \urcorner \rho \in \mathrm{FTy}_0(\mathbf{y}\, I)$ to be given by the sets $(\ulcorner A \urcorner \rho) f = A(\rho f)$ and likewise for restrictions and composition structure. These operations satisfy $\mathsf{El} \ulcorner A \urcorner = A$ and $\ulcorner \mathsf{El}\, T \urcorner = T$.

8.3 Interpretation of the syntax

Following [22], we define a partial interpretation function from raw syntax to the CwF with fibrant types given in the previous section.

To interpret the universe rules à la Russell we assume two Grothendieck universes in the underlying set theory, say *tiny* and *small* sets. So that any tiny set is small, and the set of tiny sets is small. For a cubical set X we define $\mathrm{FTy}_0(X)$ and $\mathrm{FTy}_1(X)$ as in the previous section, now referring to tiny and small sets, respectively. We get semantic universes $\mathtt{U}_i(I) = \mathrm{FTy}_i(\mathbf{y}\, I)$ for $i = 0, 1$; we identify those with their lifts to types. As noted above, these lifts carry a composition structure, and thus are fibrant. We also have $\mathtt{U}_0 \subseteq \mathtt{U}_1$ and thus $\mathrm{Ter}(X; \mathtt{U}_0) \subseteq \mathrm{Ter}(X; \mathtt{U}_1)$. Note that coding and decoding are, as set-theoretic operations, the same for both universes. We get that $\ulcorner \mathtt{U}_0 \urcorner \in \mathrm{Ter}(X; \mathtt{U}_1)$ which will serve as the interpretation of U.

In what follows, we define a partial interpretation function of raw syntax: $[\![\Gamma]\!]$, $[\![\Gamma; t]\!]$, and $[\![\Delta; \sigma]\!]$ by recursion on the raw syntax. Since we want to interpret a universe à la Russell we cannot assume terms and types to have different syntactic categories. The definition is given below and should be read such that the interpretation is defined whenever all interpretations on the right hand sides are defined *and* make sense; so, e.g., for $[\![\Gamma]\!].\, \mathsf{El}\, [\![\Gamma; A]\!]$ below, we require that $[\![\Gamma]\!]$ is defined and a cubical set, $[\![\Gamma; A]\!]$ is defined, and $\mathsf{El}\, [\![\Gamma; A]\!] \in \mathrm{FTy}([\![\Gamma]\!])$. The interpretation for raw contexts is given by:

$$
\begin{aligned}
&[\![()]\!] = \top & &[\![\Gamma, x : A]\!] = [\![\Gamma]\!].\mathsf{El}\, [\![\Gamma; A]\!] & &\text{if } x \notin \mathrm{dom}(\Gamma)\\
&[\![\Gamma, \varphi]\!] = [\![\Gamma]\!], [\![\Gamma; \varphi]\!] & &[\![\Gamma, i : \mathbb{I}]\!] = [\![\Gamma]\!].\mathbb{I} & &\text{if } i \notin \mathrm{dom}(\Gamma)
\end{aligned}
$$

where $\top$ is a terminal cubical set and in the last equation $\mathbb{I}$ is considered as an element of $\mathrm{Ty}([\![\Gamma]\!])$. When defining $[\![\Gamma; t]\!]$ we require that $[\![\Gamma]\!]$ is defined and a cubical set; then $[\![\Gamma; t]\!]$ is a (partial) family of sets $[\![\Gamma; t]\!](I, \rho)$ for $I \in \mathcal{C}$ and $\rho \in [\![\Gamma]\!](I)$ (leaving I implicit in the definition). We define:

$$
\begin{aligned}
[\![\Gamma; \mathsf{U}]\!] &= \ulcorner \mathtt{U}_0 \urcorner \in \mathrm{Ter}([\![\Gamma]\!]; \mathtt{U}_1)\\
[\![\Gamma; \mathsf{N}]\!] &= \ulcorner \mathtt{N} \urcorner \in \mathrm{Ter}([\![\Gamma]\!]; \mathtt{U}_0)\\
[\![\Gamma; (x : A) \to B]\!] &= \ulcorner \Pi_{[\![\Gamma]\!]}(\mathsf{El}\, [\![\Gamma; A]\!], \mathsf{El}\, [\![\Gamma, x : A; B]\!]) \urcorner\\
[\![\Gamma; (x : A) \times B]\!] &= \ulcorner \Sigma_{[\![\Gamma]\!]}(\mathsf{El}\, [\![\Gamma; A]\!], \mathsf{El}\, [\![\Gamma, x : A; B]\!]) \urcorner\\
[\![\Gamma; \mathsf{Path}\ A\ a\ b]\!] &= \ulcorner \mathtt{Path}^{[\![\Gamma]\!]}_{\mathsf{El}\, [\![\Gamma; A]\!]}([\![\Gamma; a]\!], [\![\Gamma; b]\!]) \urcorner\\
[\![\Gamma; \mathsf{Glue}\ [\varphi \mapsto (T, f)]\ A]\!] &= \ulcorner \mathtt{Glue}_{[\![\Gamma]\!]}([\![\Gamma; \varphi]\!], \mathsf{El}\, [\![\Gamma, \varphi; T]\!], \mathsf{El}\, [\![\Gamma; A]\!], [\![\Gamma, \varphi; f]\!]) \urcorner\\
[\![\Gamma; \lambda x : A.t]\!] &= \lambda_{[\![\Gamma]\!]; \mathsf{El}\, [\![\Gamma; A]\!]}([\![\Gamma, x : A; t]\!])\\
[\![\Gamma; t\ u]\!] &= \mathtt{app}([\![\Gamma; t]\!], [\![\Gamma; u]\!])\\
[\![\Gamma; \langle i \rangle\ t]\!] &= \langle\, \rangle_{[\![\Gamma]\!]} [\![\Gamma, i \colon \mathbb{I}; t]\!]\\
[\![\Gamma; t\ r]\!] &= [\![\Gamma; t]\!] [\![\Gamma; r]\!]
\end{aligned}
$$

where for path application, juxtaposition on the right hand side is semantic path application. In the case of a bound variable, we assume that x (respectively i) is a *chosen* variable fresh for Γ; if this is not possible the expression is undefined. Moreover, all type formers should be read as those on fibrant types, i.e., also defining the composition structure. In the case of `Glue`, it is understood that the function part, i.e., the fourth argument of `Glue` in Definition 15 is $\mathsf{p} \circ [\![\Gamma, \varphi; f]\!]$ and the required (by Theorem 17) equivalence structure is to be extracted from $\mathsf{q} \circ [\![\Gamma, \varphi; f]\!]$ as in Section 5.1. In virtue of Remark 8.2 we don't need type annotations to interpret applications. Note that coding and decoding tacitly refer to $[\![\Gamma]\!]$ as well. For the rest of the raw terms we also assume we are given $\rho \in [\![\Gamma]\!](I)$. Variables are interpreted by:

$$[\![\Gamma, x : A; x]\!]\rho = \mathsf{q}(\rho) \qquad [\![\Gamma, x : A; y]\!]\rho = [\![\Gamma; y]\!](\mathsf{p}(\rho)) \qquad [\![\Gamma, \varphi; y]\!]\rho = [\![\Gamma; y]\!]\rho$$

These should also be read to include the case when x or y are name variables; if x is a name variable, we require A to be $\mathbb{I}$. The interpretations of $[\![\Gamma; r]\!]\rho$ where r is not a name and $[\![\Gamma; \varphi]\!]\rho$ follow inductively as elements of $\mathbb{I}$ and $\mathbb{F}$, respectively.

Constants for dependent sums are interpreted by:

$$[\![\Gamma; (t, u)]\!]\rho = ([\![\Gamma; t]\!]\rho, [\![\Gamma; u]\!]\rho) \qquad [\![\Gamma; t.1]\!]\rho = \mathsf{p}([\![\Gamma; t]\!]\rho) \qquad [\![\Gamma; t.2]\!]\rho = \mathsf{q}([\![\Gamma; t]\!]\rho)$$

Likewise, constants for N will be interpreted by their semantic analogues (omitted). The interpretations for the constants related to glueing are

$$[\![\Gamma; \mathsf{glue}\, [\varphi \mapsto t]\, u]\!]\rho = \mathtt{glue}([\![\Gamma; \varphi]\!]\rho, [\![\Gamma, \varphi; t]\!]\hat{\rho}, [\![\Gamma; u]\!]\rho)$$
$$[\![\Gamma; \mathsf{unglue}\, [\varphi \mapsto (T, f)]\, u]\!]\rho = \mathtt{unglue}([\![\Gamma; \varphi]\!], \mathsf{El}\, [\![\Gamma; T]\!], \mathsf{p} \circ [\![\Gamma; f]\!])(\rho, [\![\Gamma; u]\!]\rho)$$

where $[\![\Gamma, \varphi; t]\!]\hat{\rho}$ is the family assigning $[\![\Gamma, \varphi; t]\!](\rho f)$ to $J \in \mathcal{C}$ and $f\colon J \to I$ (and ρf refers to the restriction given by $[\![\Gamma]\!]$ which is assumed to be a cubical set). Partial elements are interpreted by

$$[\![\Gamma; [\ \varphi_1\ t_1, \ldots, \varphi_n\ t_n\ ]]\!]\rho = [\![\Gamma, \varphi_i; u_i]\!]\rho \qquad \text{if } [\![\Gamma; \varphi_i]\!]\rho = 1_{\mathbb{F}},$$

where for this to be defined we additionally assume that all $[\![\Gamma, \varphi_i; u_i]\!]$ are defined and $[\![\Gamma, \varphi_i; u_i]\!]\rho' = [\![\Gamma, \varphi_j; u_j]\!]\rho'$ for each $\rho' \in [\![\Gamma]\!](I)$ with $[\![\Gamma; \phi_i \wedge \phi_j]\!]\rho' = 1_{\mathbb{F}}$.

Finally, the interpretation of composition is given by

$$[\![\Gamma; \mathsf{comp}^i\ A\ [\varphi \mapsto u]\ a_0]\!]\rho = \mathtt{comp}_{\mathsf{El}\,[\![\Gamma, i:\mathbb{I}; A]\!]}(I, j, \rho', [\![\Gamma; \varphi]\!]\rho, [\![\Gamma, \varphi, i\colon \mathbb{I}; u]\!]\rho', [\![\Gamma; a_0]\!]\rho)$$

if $i \notin \mathrm{dom}(\Gamma)$, and where j is fresh and $\rho' = (\rho \mathrm{s}_j, i = j)$ with $\mathrm{s}_j\colon I, j \to I$ induced from the inclusion $I \subseteq I, j$.

The interpretation of raw substitutions $[\![\Delta; \sigma]\!]$ is a (partial) family of sets $[\![\Delta; \sigma]\!](I, \rho)$ for $I \in \mathcal{C}$ and $\rho \in [\![\Delta]\!](I)$. We set

$$[\![\Delta; ()]\!]\rho = *, \qquad [\![\Delta; (\sigma, x/t)]\!]\rho = ([\![\Delta; \sigma]\!]\rho, [\![\Delta; t]\!]\rho) \quad \text{if } x \notin \mathrm{dom}(\sigma),$$

where $*$ is the unique element of $\top(I)$. This concludes the definition of the interpretation of syntax.

In the following α stands for either a raw term or raw substitution. In the latter case, $\alpha\sigma$ denotes composition of substitutions.

▶ **Lemma 19.** *Let Γ' be like Γ but with some φ's inserted, and assume both $[\![\Gamma]\!]$ and $[\![\Gamma']\!]$ are defined; then:*

1. *$[\![\Gamma']\!]$ is a sub-cubical set of $[\![\Gamma]\!]$;*
2. *if $[\![\Gamma;\alpha]\!]$ is defined, then so is $[\![\Gamma';\alpha]\!]$ and they agree on $[\![\Gamma']\!]$.*

▶ **Lemma 20** (Weakening). *Let $[\![\Gamma]\!]$ be defined.*

1. *If $[\![\Gamma, x : A, \Delta]\!]$ is defined, then so is $[\![\Gamma, x : A, \Delta; x]\!]$ which is moreover the projection to the x-part.*[5]
2. *If $[\![\Gamma, \Delta]\!]$ is defined, then so is $[\![\Gamma, \Delta; \mathrm{id}_\Gamma]\!]$ which is moreover the projection to the Γ-part.*
3. *If $[\![\Gamma, \Delta]\!]$, $[\![\Gamma; \alpha]\!]$ are defined and the variables in Δ are fresh for α, then $[\![\Gamma, \Delta; \alpha]\!]$ is defined and for $\rho \in [\![\Gamma, \Delta]\!](I)$:*

$$[\![\Gamma;\alpha]\!]([\![\Gamma,\Delta;\mathrm{id}_\Gamma]\!]\rho) = [\![\Gamma,\Delta;\alpha]\!]\rho$$

▶ **Lemma 21** (Substitution). *For $[\![\Gamma]\!]$,$[\![\Delta]\!]$, $[\![\Delta;\sigma]\!]$, and $[\![\Gamma;\alpha]\!]$ defined with $\mathrm{dom}(\Gamma) = \mathrm{dom}(\sigma)$ (as lists), also $[\![\Delta;\alpha\sigma]\!]$ is defined and for $\rho \in [\![\Delta]\!](I)$:*

$$[\![\Gamma;\alpha]\!]([\![\Delta;\sigma]\!]\rho) = [\![\Delta;\alpha\sigma]\!]\rho$$

▶ **Lemma 22.** *If $[\![\Gamma]\!]$ is defined and a cubical set, and $[\![\Gamma;\alpha]\!]$ is defined, then $([\![\Gamma;\alpha]\!]\rho)f = [\![\Gamma;\alpha]\!](\rho f)$.*

To state the next theorem let us set $[\![\Gamma;\mathbb{I}]\!] = \ulcorner\mathbb{I}\urcorner$ and $[\![\Gamma;\mathbb{F}]\!] = \ulcorner\mathbb{F}\urcorner$ as elements of $\mathrm{Ty}_0([\![\Gamma]\!])$.

▶ **Theorem 23** (Soundness). *We have the following implications, and all occurrences of $[\![-]\!]$ in the conclusions are defined. Where in (3) and (5) we allow A to be $\mathbb{I}$ or $\mathbb{F}$.*

1. *if $\Gamma \vdash$, then $[\![\Gamma]\!]$ is a cubical set;*
2. *if $\Gamma \vdash A$, then $[\![\Gamma;A]\!] \in \mathrm{Ter}([\![\Gamma]\!];\mathtt{U}_1)$;*
3. *if $\Gamma \vdash t : A$, then $[\![\Gamma;t]\!] \in \mathrm{Ter}([\![\Gamma]\!];\mathsf{El}\,[\![\Gamma;A]\!])$;*
4. *if $\Gamma \vdash A = B$, then $[\![\Gamma;A]\!] = [\![\Gamma;B]\!]$;*
5. *if $\Gamma \vdash a = b : A$, then $[\![\Gamma;a]\!] = [\![\Gamma;b]\!]$;*
6. *if $\Gamma \vdash \sigma : \Delta$, then $[\![\Gamma;\sigma]\!]$ restricts to a natural transformation $[\![\Gamma]\!] \to [\![\Delta]\!]$.*

9 Extensions: identity types and higher inductive types

In this section we consider possible extensions to cubical type theory. The first is an identity type defined using path types whose elimination principle holds as a judgmental equality. The second are two examples of higher inductive types.

9.1 Identity types

We can use the path type to represent equalities. Using the composition operation, we can indeed build a substitution function $P(a) \to P(b)$ from any path between a and b. However, since we don't have in general the judgmental equality $\mathsf{transp}^i\ A\ a_0 = a_0$ if A is independent of i (which is an equality that we cannot expect geometrically in general, as shown in Appendix C), this substitution function does not need to be the constant function when the path is constant. This means that, as in the previous model [6, 13], we don't get an interpretation of Martin-Löf identity type [17] with the standard judgmental equalities.

However, we can define another type which *does* give an interpretation of this identity type following an idea of Andrew Swan.

[5] E.g., if Γ is $y : B, z : C$, the projection to the x-part maps $(b,(c,(a,\delta)))$ to a, and the projection to the Γ-part maps $(b,(c,\delta))$ to (b,c).

Identity types

We define a type $\mathsf{Id}\ A\ a_0\ a_1$ with the introduction rule

$$\frac{\Gamma \vdash \omega : \mathsf{Path}\ A\ a_0\ a_1[\varphi \mapsto \langle i\rangle\ a_0]}{\Gamma \vdash (\omega, \varphi) : \mathsf{Id}\ A\ a_0\ a_1}$$

and $\mathsf{r}(a) = (\langle j\rangle\ a, 1_{\mathbb{F}}) : \mathsf{Id}\ A\ a\ a$.

Given $\Gamma \vdash \alpha = (\omega, \varphi) : \mathsf{Id}\ A\ a\ x$ we define $\Gamma, i : \mathbb{I} \vdash \alpha^*(i) : \mathsf{Id}\ A\ a\ (\alpha\ i)$ by

$$\alpha^*(i) = (\langle j\rangle\ \omega\ (i \wedge j), \varphi \vee (i = 0))$$

This is well defined since $\Gamma, i : \mathbb{I}, (i = 0) \vdash \langle j\rangle\ \omega\ (i \wedge j) = \langle j\rangle\ a$ and $\Gamma, i : \mathbb{I}, \varphi \vdash \langle j\rangle\ \omega\ (i \wedge j) = \langle j\rangle\ a$.

If we have $\Gamma, x : A, \alpha : \mathsf{Id}\ A\ a\ x \vdash C$, $\Gamma \vdash b : A$, $\Gamma \vdash \beta : \mathsf{Id}\ A\ a\ b$, and $\Gamma \vdash d : C(a, \mathsf{r}(a))$ we define

$$J\ b\ \beta\ d = \mathsf{comp}^i\ C(\omega\ i, \beta^*(i))\ [\varphi \mapsto d]\ d : C(b, \beta)$$

where $\beta = (\omega, \varphi)$ and we have $J\ a\ \mathsf{r}(a)\ d = d$ as desired.

If $i : \mathbb{I} \vdash \mathsf{Id}\ A\ a\ b$ and $p_0 = (\omega_0, \psi_0) : \mathsf{Id}\ A(i0)\ a(i0)\ b(i0)$ and $\varphi, i : \mathbb{I} \vdash q = (\omega, \psi) : \mathsf{Id}\ A\ a\ b$ such that $\varphi \vdash q(i0) = p_0$ we define $\mathsf{comp}^i\ (\mathsf{Id}\ A\ a\ b)\ [\varphi \mapsto q]\ p_0$ to be $(\gamma, \varphi \wedge \psi(i1))$ where

$$\gamma = \langle j\rangle\ \mathsf{comp}^i\ A\ [\varphi \mapsto \omega\ j, (j = 0) \mapsto a, (j = 1) \mapsto b]\ (\omega_0\ j)$$

It can then be shown that the types $\mathsf{Id}\ A\ a\ b$ and $\mathsf{Path}\ A\ a\ b$ are (path)-equivalent. In particular, a type is (path)-contractible if, and only if, it is (Id)-contractible. Corollary 11, which is a statement about the Path type, holds hence as well for the Id type.

Cofibration-trivial fibration factorization

The same idea can be used to factorize an arbitrary map $f : A \to B$ into a cofibration followed by a trivial fibration. We define a "trivial fibration" to be a first projection from a total space of a contractible family of types and a "cofibration" to be a map that has the left lifting property against any trivial fibration. For this we define, for $b : B$, the type $T_f(b)$ to be the type of elements $\varphi \mapsto a$ with $\varphi \vdash a : A$ and $\varphi \vdash f\ a = b : B$.

▶ **Theorem 24.** *The type $T_f(b)$ is contractible and the map*

$$A \to (b : B) \times T_f(b), \qquad a \longmapsto (f\ a, [1_{\mathbb{F}} \mapsto a])$$

is a cofibration.

The definition of the identity type can be seen as a special case of this, if we take the B the type of paths in A and for f the constant path function.

9.2 Higher inductive types

In this section we consider the extension of cubical type theory with two different higher inductive types: spheres and propositional truncation. The presentation in this section is syntactical, but it can be directly translated into semantic definitions.

Extension to dependent path types

In order to formulate the elimination rules for higher inductive types, we need to extend the path type to *dependent path type*, which is described by the following rules. If $i : \mathbb{I} \vdash A$ and $\vdash a_0 : A(i0)$, $a_1 : A(i1)$, then $\vdash \mathsf{Path}^i\ A\ a_0\ a_1$. The introduction rule is that $\vdash \langle i \rangle\ t : \mathsf{Path}^i\ A\ t(i0)\ t(i1)$ if $i : \mathbb{I} \vdash t : A$. The elimination rule is $\vdash p\ r : A(i/r)$ if $\vdash p : \mathsf{Path}^i\ A\ a_0\ a_1$ with equalities $p\ 0 = a_0 : A(i0)$ and $p\ 1 = a_1 : A(i1)$.

Spheres

We define the circle, S^1, by the rules:

$$\frac{\Gamma \vdash}{\Gamma \vdash \mathsf{S}^1} \qquad \frac{\Gamma \vdash}{\Gamma \vdash \mathsf{base} : \mathsf{S}^1} \qquad \frac{\Gamma \vdash r : \mathbb{I}}{\Gamma \vdash \mathsf{loop}(r) : \mathsf{S}^1}$$

with the equalities $\mathsf{loop}(0) = \mathsf{loop}(1) = \mathsf{base}$.

Since we want to represent the *free* type with one base point and a loop, we add composition as a *constructor* operation hcomp^i:

$$\frac{\Gamma, \varphi, i : \mathbb{I} \vdash u : \mathsf{S}^1 \qquad \Gamma \vdash u_0 : \mathsf{S}^1[\varphi \mapsto u(i0)]}{\Gamma \vdash \mathsf{hcomp}^i\ [\varphi \mapsto u]\ u_0 : \mathsf{S}^1}$$

with the equality $\mathsf{hcomp}^i\ [1_{\mathbb{F}} \mapsto u]\ u_0 = u(i1)$.

Given a dependent type $x : \mathsf{S}^1 \vdash A$ and $a : A(x/\mathsf{base})$ and $l : \mathsf{Path}^i\ A(x/\mathsf{loop}(i))\ a\ a$ we can define a function $g : (x : \mathsf{S}^1) \to A$ by the equations[6] $g\ \mathsf{base} = a$ and $g\ \mathsf{loop}(r) = l\ r$ and

$$g\ (\mathsf{hcomp}^i\ [\varphi \mapsto u]\ u_0) = \mathsf{comp}^i\ A(x/v)\ [\varphi \mapsto g\ u]\ (g\ u_0)$$

where $v = \mathsf{fill}^i\ \mathsf{S}^1\ [\varphi \mapsto u]\ u_0 = \mathsf{hcomp}^j\ [\varphi \mapsto u(i/i \wedge j), (i = 0) \mapsto u_0]\ u_0$.

This definition is non ambiguous since $l\ 0 = l\ 1 = a$.

We have a similar definition for S^n taking as constructors base and $\mathsf{loop}(r_1, \ldots, r_n)$.

Propositional truncation

We define the propositional truncation, $\mathsf{inh}\ A$, of a type A by the rules:

$$\frac{\Gamma \vdash A}{\Gamma \vdash \mathsf{inh}\ A} \qquad \frac{\Gamma \vdash a : A}{\Gamma \vdash \mathsf{inc}\ a : \mathsf{inh}\ A} \qquad \frac{\Gamma \vdash u_0 : \mathsf{inh}\ A \qquad \Gamma \vdash u_1 : \mathsf{inh}\ A \qquad \Gamma \vdash r : \mathbb{I}}{\Gamma \vdash \mathsf{squash}(u_0, u_1, r) : \mathsf{inh}\ A}$$

with the equalities $\mathsf{squash}(u_0, u_1, 0) = u_0$ and $\mathsf{squash}(u_0, u_1, 1) = u_1$.

As before, we add composition as a constructor, but only in the form[7]

$$\frac{\Gamma, \varphi, i : \mathbb{I} \vdash u : \mathsf{inh}\ A \qquad \Gamma \vdash u_0 : \mathsf{inh}\ A[\varphi \mapsto u(i0)]}{\Gamma \vdash \mathsf{hcomp}^i\ [\varphi \mapsto u]\ u_0 : \mathsf{inh}\ A}$$

with the equality $\mathsf{hcomp}^i\ [1_{\mathbb{F}} \mapsto u]\ u_0 = u(i1)$.

This provides only a definition of $\mathsf{comp}^i\ (\mathsf{inh}\ A)\ [\varphi \mapsto u]\ u_0$ in the case where A is independent of i, and we have to explain how to define the general case.

---

[6] For the equation $g\ \mathsf{loop}(r) = l\ r$, it may be that l and r are dependent on the same name i, and we could not have followed this definition in the framework of [6].

[7] This restriction on the constructor is essential for the justification of the elimination rule below.

In order to do this, we define first two operations

$$\frac{\Gamma, i : \mathbb{I} \vdash A \qquad \Gamma \vdash u_0 : \mathsf{inh}\ A(i0)}{\Gamma \vdash \mathsf{transp}\ u_0 : \mathsf{inh}\ A(i1)} \qquad \frac{\Gamma, i : \mathbb{I} \vdash A \qquad \Gamma, i : \mathbb{I} \vdash u : \mathsf{inh}\ A}{\Gamma \vdash \mathsf{squeeze}^i\ u : \mathsf{Path}\ (\mathsf{inh}\ A(i1))\ (\mathsf{transp}\ u(i0))\ u(i1)}$$

by the equations

$$\begin{array}{lcl}
\mathsf{transp}\ (\mathsf{inc}\ a) & = & \mathsf{inc}\ (\mathsf{comp}^i\ A\ []\ a) \\
\mathsf{transp}\ (\mathsf{squash}(u_0, u_1, r)) & = & \mathsf{squash}(\mathsf{transp}\ u_0, \mathsf{transp}\ u_1, r) \\
\mathsf{transp}\ (\mathsf{hcomp}^j\ [\varphi \mapsto u]\ u_0) & = & \mathsf{hcomp}^j\ [\varphi \mapsto \mathsf{transp}\ u]\ (\mathsf{transp}\ u_0) \\
\\
\mathsf{squeeze}^i\ (\mathsf{inc}\ a) & = & \langle i \rangle\ \mathsf{inc}\ (\mathsf{comp}^j\ A(i \vee j)\ [(i = 1) \mapsto a(i1)]\ a) \\
\mathsf{squeeze}^i\ (\mathsf{squash}(u_0, u_1, r)) & = & \langle k \rangle\ \mathsf{squash}(\mathsf{squeeze}^i\ u_0\ k, \mathsf{squeeze}^i\ u_1\ k, r(i/k)) \\
\mathsf{squeeze}^i\ (\mathsf{hcomp}^j\ [\varphi \mapsto u]\ v) & = & \langle k \rangle\ \mathsf{hcomp}^j\ S\ (\mathsf{squeeze}^i\ v\ k)
\end{array}$$

where S is the system

$$[\delta \mapsto \mathsf{squeeze}^i\ u\ k,\ \ \varphi(i/k) \wedge (k = 0) \mapsto \mathsf{transp}\ u(i0),\ \ \varphi(i/k) \wedge (k = 1) \mapsto u(i1)]$$

and $\delta = \forall i.\varphi$, using Lemma 2.

Using these operations, we can define the general composition

$$\frac{\Gamma, i : \mathbb{I} \vdash A \qquad \Gamma, \varphi, i : \mathbb{I} \vdash u : \mathsf{inh}\ A \qquad \Gamma \vdash u_0 : \mathsf{inh}\ A(i0)[\varphi \mapsto u(i0)]}{\Gamma \vdash \mathsf{comp}^i\ (\mathsf{inh}\ A)\ [\varphi \mapsto u]\ u_0 : \mathsf{inh}\ A(i1)[\varphi \mapsto u(i1)]}$$

by $\Gamma \vdash \mathsf{comp}^i\ (\mathsf{inh}\ A)\ [\varphi \mapsto u]\ u_0 = \mathsf{hcomp}^j\ [\varphi \mapsto \mathsf{squeeze}^i\ u\ j]\ (\mathsf{transp}\ u_0) : \mathsf{inh}\ A(i1)$.

Given $\Gamma \vdash B$ and $\Gamma \vdash q : (x\ y : B) \to \mathsf{Path}\ B\ x\ y$ and $f : A \to B$ we define $g : \mathsf{inh}\ A \to B$ by the equations

$$\begin{array}{lcl}
g\ (\mathsf{inc}\ a) & = & f\ a \\
g\ (\mathsf{squash}(u_0, u_1, r)) & = & q\ (g\ u_0)\ (g\ u_1)\ r \\
g\ (\mathsf{hcomp}^j\ [\varphi \mapsto u]\ u_0) & = & \mathsf{comp}^j\ B\ [\varphi \mapsto g\ u]\ (g\ u_0)
\end{array}$$

10 Related and future work

Cubical ideas have proved useful to reason about equality in homotopy type theory [16]. In cubical type theory these techniques could be simplified as there are new judgmental equalities and better notations for manipulating higher dimensional cubes. Indeed some simple experiments using the HASKELL implementation have shown that we can simplify some constructions in synthetic homotopy theory.[8]

Other approaches to extending intensional type theory with extensionality principles can be found in [2, 21]. These approaches have close connections to techniques for internalizing parametricity in type theory [5]. Further, nominal extensions to λ-calculus and semantical ideas related to the ones presented in this paper have recently also proved useful for justifying type theory with internalized parametricity [4].

The paper [11] provides a general framework for analyzing the uniformity condition, which applies to simplicial and cubical sets.

[8] For details see: `https://github.com/mortberg/cubicaltt/tree/master/examples`

In Section 4.4 we show how to define Kan filling from composition. The semantics of this, which follows directly the definition in Section 4.4, has been formally verified in NuPrl by Mark Bickford.[9]

Following the usual reducibility method, we expect it to be possible to adapt our presheaf semantics to a proof of normalization and decidability of type checking. We end the paper with a list of open problems and conjectures:

1. Show that any cubical group has a uniform Kan composition operation.
2. Extend the system with resizing rules and show normalization.
3. Extend the semantics of identity types to the semantics of inductive families.
4. Give a general syntax and semantics of higher inductive types.

Acknowledgements The authors would like to thank Martín Escardó, Georges Gonthier, Dan Grayson, Peter Hancock, Dan Licata, Peter LeFanu Lumsdaine, Christian Sattler, Andrew Swan, and Vladimir Voevodsky for many interesting discussions and remarks.

References

1 P. Aczel. On Relating Type Theories and Set Theories. In T. Altenkirch, B. Reus, and W. Naraschewski, editors, *Types for Proofs and Programs*, volume 1657 of *Lecture Notes in Computer Science*, pages 1–18. Springer Verlag, Berlin, Heidelberg, New York, 1999.

2 T. Altenkirch. Extensional Equality in Intensional Type Theory. In *14th Annual IEEE Symposium on Logic in Computer Science*, pages 412–420, 1999.

3 R. Balbes and P. Dwinger. *Distributive Lattices.* Abstract Space Publishing, 2011.

4 J.-P. Bernardy, T. Coquand, and G. Moulin. A Presheaf Model of Parametric Type Theory. Preprint, available online, May 2015.

5 J.-P. Bernardy and G. Moulin. Type-theory in Color. *SIGPLAN Not.*, 48(9):61–72, September 2013.

6 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)*, volume 26 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 107–128, 2014.

7 R. Brown, P.J. Higgins, and R. Sivera. *Nonabelian Algebraic Topology: Filtered Spaces, Crossed Complexes, Cubical Homotopy Groupoids*, volume 15 of *EMS tracts in mathematics*. European Mathematical Society, 2011.

8 D.-C. Cisinski. Univalent universes for elegant models of homotopy types. *arXiv:1406.0058*, May 2014. Preprint.

9 P. Dybjer. Internal Type Theory. In *Lecture Notes in Computer Science*, pages 120–134. Springer Verlag, Berlin, Heidelberg, New York, 1996.

10 M. Fourman and D. Scott. Sheaves and logic. In M. Fourman, C. Mulvey, and D. Scott, editors, *Applications of Sheaves*, volume 753 of *Lecture Notes in Mathematics*, pages 302–401. Springer Berlin Heidelberg, 1979.

11 N. Gambino and C. Sattler. Uniform Fibrations and the Frobenius Condition. *arXiv:1510.00669*, October 2015. Preprint.

12 M. Hofmann. Syntax and semantics of dependent types. In A.M. Pitts and P. Dybjer, editors, *Semantics and logics of computation*, volume 14 of *Publ. Newton Inst.*, pages 79–130. Cambridge University Press, Cambridge, 1997.

[9] For details see: `http://www.nuprl.org/wip/Mathematics/cubical!type!theory/`

13 S. Huber. *A Model of Type Theory in Cubical Sets*. Licentiate thesis, University of Gothenburg, May 2015.

14 D. M. Kan. Abstract homotopy. I. *Proceedings of the National Academy of Sciences of the United States of America*, 41(12):1092–1096, 1955.

15 C. Kapulkin, P. LeFanu Lumsdaine, and V. Voevodsky. The Simplicial Model of Univalent Foundations. *arXiv:1211.2851*, November 2012. Preprint.

16 D. R. Licata and G. Brunerie. A Cubical Approach to Synthetic Homotopy Theory. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 92–103, 2015.

17 P. Martin-Löf. An intuitionistic theory of types: predicative part. In *Logic Colloquium '73 (Bristol, 1973)*, pages 73–118. Studies in Logic and the Foundations of Mathematics, Vol. 80. North-Holland, Amsterdam, 1975.

18 P. Martin-Löf. An intuitionistic theory of types. In G. Sambin and J. M. Smith, editors, *Twenty-five years of constructive type theory (Venice, 1995)*, volume 36 of *Oxford Logic Guides*, pages 127–172. Oxford University Press, 1998.

19 A. M. Pitts. *Nominal Sets: Names and Symmetry in Computer Science*. Cambridge University Press, New York, NY, USA, 2013.

20 A. M. Pitts. Nominal Presentation of Cubical Sets Models of Type Theory. In H. Herbelin, P. Letouzey, and M. Sozeau, editors, *20th International Conference on Types for Proofs and Programs (TYPES 2014)*, volume 39 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 202–220. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2015.

21 A. Polonsky. Extensionality of lambda-*. In H. Herbelin, P. Letouzey, and M. Sozeau, editors, *20th International Conference on Types for Proofs and Programs, TYPES 2014*, volume 39 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 221–250. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2014.

22 T. Streicher. *Semantics of Type Theory*. Progress in Theoretical Computer Science. Birkhäuser Basel, 1991.

23 A. Swan. An Algebraic Weak Factorisation System on 01-Substitution Sets: A Constructive Proof. *arXiv:1409.1829*, September 2014. Preprint.

24 The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. `http://homotopytypetheory.org/book`, Institute for Advanced Study, 2013.

25 V. Voevodsky. The equivalence axiom and univalent models of type theory. (Talk at CMU on February 4, 2010). *arXiv:1402.5556*, February 2014. Preprint.

A Details of composition for glueing

We build the element $\Gamma \vdash b_1 = \mathsf{comp}^i\ B\ [\psi \mapsto b]\ b_0 : (\mathsf{Glue}\ [\varphi \mapsto (T, f)]\ A)(i1)$ as the element $\mathsf{glue}\ [\psi(i1) \mapsto\ t_1]\ a_1$ where

$$\Gamma, \varphi(i1) \vdash t_1 : T(i1)[\psi \mapsto b(i1)]$$
$$\Gamma \vdash a_1 : A(i1)[\varphi(i1) \mapsto f(i1)\ t_1, \psi \mapsto (\mathsf{unglue}\ b)(i1)]$$

As intermediate steps, we gradually build elements that satisfy more and more of the equations that the final elements t_1 and a_1 should satisfy. The construction of these is given in five steps.

Before explaining how we can define them and why they are well defined, we illustrate the construction in the Figure 5, with $\psi = (j = 1)$ and $\varphi = (i = 0) \vee (j = 1) \vee (i = 1)$.

We pose $\delta = \forall i.\varphi$ (cf. Section 3), so that we have that δ is independent from i, and in our example $\delta = (j = 1)$ and it represents the right hand side of the picture.

1. The element $a_1' : A(i1)$ is a first approximation of a_1, but a_1' is not necessarily in the image of $f(i1)$ in $\Gamma, \varphi(i1)$;
2. the partial element $\delta \vdash t_1' : T(i1)$, which is a partial final result for $\varphi(i1) \vdash t_1$;
3. the partial path $\delta \vdash \omega$, between a_1' and the image of t_1';
4. both the final element $\varphi(i1) \vdash t_1$ and a path $\varphi(i1) \vdash \alpha$ between a_1' and $f(i1)\ t_1$;
5. finally, we build a_1 from a_1' and α.

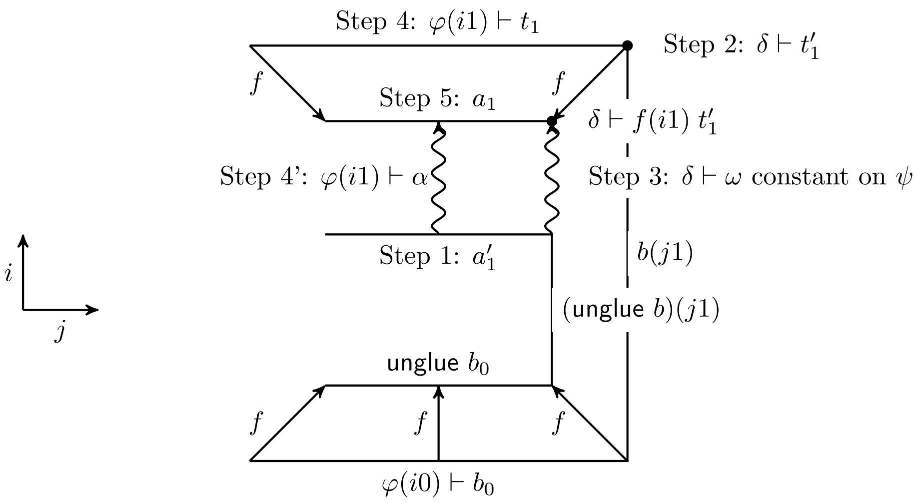


Figure 5 Composition for glueing

We define:

$$\Gamma, \psi, i : \mathbb{I} \vdash a = \mathsf{unglue}\ b : A[\varphi \mapsto f\ b]$$
$$\Gamma \vdash a_0 = \mathsf{unglue}\ b_0 : A(i0)[\varphi(i0) \mapsto f(i0)\ b_0, \psi \mapsto a(i0)]$$

Step 1 We define a_1' as the composition of a and $\mathsf{unglue}\ b_0$, in the direction i, which is well defined since $\mathsf{unglue}\ b_0 = (\mathsf{unglue}\ b)(i0)$ over the extent ψ

$$\Gamma \vdash a_1' = \mathsf{comp}^i\ A\ [\psi \mapsto\ a]\ a_0 : A(i1)[\psi \mapsto\ a(i1)] \tag{4}$$

Step 2 We also define t'_1 as the composition of b and b_0, in the direction i:

$$\Gamma, \delta \vdash t'_1 = \mathsf{comp}^i\ T\ [\psi \mapsto\ b]\ b_0 : T(i1)[\psi \mapsto\ b(i1)] \tag{5}$$

which is well defined because $\begin{cases} \Gamma, \delta, i : \mathbb{I}, \psi \vdash b : T & \text{by Lemma 3} \\ \Gamma, \delta \vdash b_0 : T(i0)[\psi \mapsto b(i0)] & \text{as } \delta \leqslant \varphi(i0) \end{cases}$

Moreover, since

$$\begin{cases} \Gamma, \delta, \psi, i : \mathbb{I} \vdash a = f\ b & \text{by } \delta \leqslant \varphi \\ \Gamma, \delta \vdash\ a_0 = f(i0)\ b_0 & \text{by } \delta \leqslant \varphi(i0) \end{cases}$$

we can re-express a'_1 on the extent δ

$$\Gamma, \delta \vdash a'_1 = \mathsf{comp}^i\ A\ [\psi \mapsto f\ b]\ \ (f(i0)\ b_0)$$

Step 3 We can hence find a path ω connecting a'_1 and $f(i1)\ t'_1$ in Γ, δ using Lemma 6:

$$\Gamma, \delta \vdash \omega = \mathsf{pres}^i\ f\ [\psi \mapsto b]\ b_0 : (\mathsf{Path}\ A(i1)\ a'_1\ \ (f(i1)\ t'_1))\ [\psi \mapsto \langle j \rangle\ a(i1)]$$

Picking a fresh name j, we have

$$\Gamma, \delta, j : \mathbb{I} \vdash \omega\ j : A(i1)[(j0) \mapsto a'_1, (j1) \mapsto f(i1)\ t'_1, \psi \mapsto a(i1)] \tag{6}$$

Step 4 Now we define the final element t_1 as the inverse image of a'_1 by $f(i1)$, together with the path α between a'_1 and $f(i1)\ t_1$, in $\Gamma, \varphi(i1) \vdash$, using Lemma 7:

$$\Gamma, \varphi(i1) \vdash (t_1, \alpha) = \mathsf{equiv}\ f(i1)\ [\delta \mapsto\ (t'_1, \omega), \psi \mapsto\ (b(i1), \langle j \rangle\ a'_1)]\ a'_1$$

with $\begin{cases} \Gamma, \varphi(i1) \vdash t_1 : T(i1)[\delta \mapsto\ t'_1, \psi \mapsto\ b(i1)] \\ \Gamma, \varphi(i1) \vdash \alpha : (\mathsf{Path}\ A(i1)\ a'_1\ \ (f(i1)\ t_1))\ [\delta \mapsto \omega, \psi \mapsto\ \langle j \rangle\ a'_1)] \end{cases}$

These are well defined because the two systems in δ and ψ are compatible:

$$\begin{cases} \Gamma, \delta, \psi \vdash t'_1 = b(i1) & \text{by (5)} \\ \Gamma, \delta, \psi \vdash \omega = \langle j \rangle\ a'_1 & \text{by (6) and (4)} \end{cases}$$

Picking a fresh name j, we have

$$\Gamma, \varphi(i1), j : \mathbb{I} \vdash \alpha\ j : A(i1)[(j0) \mapsto a'_1,\ (j1) \mapsto f(i1)\ t_1,\ \delta \mapsto\ a'_1,\ \psi \mapsto a(i1)] \tag{7}$$

Step 5 Finally, we define a_1 by composition of α and a'_1:

$$\Gamma \vdash\ a_1 := \mathsf{comp}^j\ A(i1)\ [\varphi(i1) \mapsto \alpha\ j, \psi \mapsto\ a(i1)]\ a'_1 : A(i1)[\varphi(i1) \mapsto \alpha\ 1, \psi \mapsto\ a(i1)]$$

which is well defined because $\begin{cases} \Gamma, j : \mathbb{I}, \varphi(i1), \psi \vdash \alpha\ j = a(i1) & \text{by (7)} \\ \Gamma, \varphi(i1) \vdash \alpha\ 0 = a'_1 & \text{by (7)} \\ \Gamma, \psi \vdash a(i1) = a'_1 & \text{by (4)} \end{cases}$

and since $\Gamma, \varphi(i1) \vdash \alpha\ 1 = f(i1)\ t_1$, we can re-express the type of a_1 in the following way:

$$\Gamma \vdash a_1 : A(i1)[\varphi(i1) \mapsto f(i1)\ t_1,\ \psi \mapsto a(i1)]$$

Which is exactly what we needed to build $\Gamma \vdash b_1 := \mathsf{glue}\ [\varphi(i1) \mapsto t_1]\ a_1 : B(i1)[\psi \mapsto b(i1)]$.

Finally we check that $b_1 = \mathsf{comp}^i\ T\ [\psi \mapsto\ b]\ b_0$ on δ:

$$
\begin{aligned}
b_1 &= \mathsf{glue}\ [\varphi(i1) \mapsto t_1]\ a_1 && \text{by def.}\\
&= t_1 : T(i1)[\delta \mapsto\ t_1', \psi \mapsto\ b(i1)] && \text{as } \varphi(i1) = 1_{\mathbb{F}}\\
&= t_1' && \text{as } \delta = 1_{\mathbb{F}}\\
&= \mathsf{comp}^i\ T\ [\psi \mapsto\ b]\ b_0 && \text{by def.}
\end{aligned}
$$

B Univalence from glueing

We also give an alternative proof of the univalence axiom for Path only involving the glue construction and not relying on Section 7.2.[10]

▶ **Lemma 25.** *For $\Gamma \vdash A : \mathsf{U}$, $\Gamma \vdash B : \mathsf{U}$, and an equivalence $\Gamma \vdash f : \mathsf{Equiv}\ A\ B$ we have the following constructions:*

1. $\Gamma \vdash \mathsf{eqToPath}\, f : \mathsf{Path}\, \mathsf{U}\, A\, B$*;*
2. $\Gamma \vdash \mathsf{Path}\, (A \to B)\, (\mathsf{transp}^i (\mathsf{eqToPath}\, f\, i)))\, (f.1)$ *is inhabited; and*
3. *if $f = \mathsf{equiv}^i (P\, i)$ for $\Gamma \vdash P : \mathsf{Path}\, \mathsf{U}\, A\, B$, then the following type is inhabited:*

$$\Gamma \vdash \mathsf{Path}\, (\mathsf{Path}\, \mathsf{U}\, A\, B)\, (\mathsf{eqToPath}\, (\mathsf{equiv}^i (P\, i)))\, P$$

Proof. For (1) we define

$$\mathsf{eqToPath}\, f = \langle i \rangle\ \mathsf{Glue}\, [(i = 0) \mapsto (A, f), (i = 1) \mapsto (B, \mathsf{equiv}^k B)]\, B. \tag{8}$$

Note that here $\mathsf{equiv}^k B$ is an equivalence between B and B (see Section 7.1). For (2) we have to closely look at how the composition was defined for Glue. By unfolding the definition, we see that the left hand side of the equality is equal $f.1$ composed with multiple transports in a constant type; using filling and functional extensionality, these transports can be shown to be equal to the identity; for details see the formal proof.

The term for (3) is given by:

$$
\begin{aligned}
C = \langle j \rangle\ \langle i \rangle\ \mathsf{Glue}\, [&(i = 0) \mapsto (A, \mathsf{equiv}^k (P\, k)),\\
&(i = 1) \mapsto (B, \mathsf{equiv}^k B),\\
&(j = 1) \mapsto (P\, i, \mathsf{equiv}^k (P(i \vee k)))]\\
&B
\end{aligned}
$$
◀

▶ **Corollary 26** (Univalence axiom)**.** *For the canonical map*

$$\mathsf{pathToEq} : (A\, B : \mathsf{U}) \to \mathsf{Path}\, \mathsf{U}\, A\, B \to \mathsf{Equiv}\ A\ B$$

we have that $\mathsf{pathToEq}\, A\, B$ is an equivalence for all $A : \mathsf{U}$ and $B : \mathsf{U}$.

[10] Both proofs of the univalence axiom have been formally verified inside the system using the Haskell implementation. For details see: `https://github.com/mortberg/cubicaltt/blob/master/examples/univalence.ctt`

Proof. Let us first show that the canonical map $\mathsf{pathToEq}$ is path equal to:

$$\mathsf{equiv} = \lambda A\, B : \mathsf{U}.\, \lambda P : \mathsf{Path}\,\mathsf{U}\,A\,B.\, \mathsf{equiv}^i(P\,i)$$

By function extensionality, it suffices to check this pointwise. Using path-induction, we may assume that P is reflexivity. In this case $\mathsf{pathToEq}\,A\,A\,1_A$ is the identity equivalence by definition. Because being an equivalence is a proposition, it thus suffices that the first component of $\mathsf{equiv}^i A$ is propositionally equal to the identity. By definition, this first component is given by transport (now in the constant type A) which is easily seen to be the identity using filling (see Section 4.4).

Thus it suffices to prove that $\mathsf{equiv}\,A\,B$ is an equivalence. To do so it is enough to give an inverse (see Theorems 4.2.3 and 4.2.6 of [24]). But $\mathsf{eqToPath}$ is a left inverse by Lemma 25 (3), and a right inverse by Lemma 25 (2) using that being an equivalence is a proposition. ◀

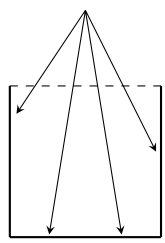
C Singular cubical sets

Recall the functor $\mathcal{C} \to \mathbf{Top}, I \mapsto [0,1]^I$ given at (1) in Section 8.1. In particular, the face maps $(ib)\colon I - i \to I$ (for $b = 0_{\mathbb{I}}$ or $1_{\mathbb{I}}$) induce the maps $(ib)\colon [0,1]^{I-i} \to [0,1]^I$ by $i(ib)u = b$ and $j(ib)u = ju$ if $j \neq i$ is in I. If ψ is in $\mathbb{F}(I)$ and u in $[0,1]^I$, then ψu is a truth value.

We assume given a family of idempotent functions $r_I : [0,1]^I \times [0,1] \to [0,1]^I \times [0,1]$ such that

1. $r_I(u,z) = (u,z)$ iff $\partial_I u = 1$ or $z = 0$, and
2. for any *strict* f in $\mathsf{Hom}(I,J)$ we have $r_J(f \times \mathrm{id})r_I = r_J(f \times \mathrm{id})$

Such a family can for instance be defined as in the following picture ("retraction from above center"). If the center has coordinate $(1/2, 2)$, then $r_I(u,z) = r_I(u',z')$ is equivalent to $(2-z')(-1+2u) = (2-z)(-1+2u')$.



Property (1) holds for the retraction defined by this picture. The property (2) can be reformulated as $r_I(u,z) = r_I(u',z') \to r_J(fu,z) = r_J(fu',z')$. It also holds in this case, since $r_I(u,z) = r_I(u',z')$ is then equivalent to $(2-z')(-1+2u) = (2-z)(-1+2u')$, which implies $(2-z')(-1+2fu) = (2-z)(-1+2fu')$ if f is strict.

Using this family, we can define for each ψ in $\mathbb{F}(I)$ an idempotent function

$$r_\psi : [0,1]^I \times [0,1] \to [0,1]^I \times [0,1]$$

having for fixed-points the element (u,z) such that $\psi u = 1$ or $z = 0$. This function r_ψ is completely characterized by the following properties

1. $r_\psi = \mathrm{id}$ if $\psi = 1$
2. $r_\psi = r_\psi r_I$ if $\psi \neq 1$

3. $r_\psi(u,z) = (u,z)$ if $z = 0$
4. $r_\psi((ib) \times \mathrm{id}) = ((ib) \times \mathrm{id})r_{\psi(ib)}$

These properties imply for instance $r_{\partial_I}(u,z) = (u,z)$ if $\partial_I u = 1$ or $z = 0$ and so they imply $r_{\partial_I} = r_I$. They also imply that $r_\psi(u,z) = (u,z)$ if $\psi u = 1$.

From these properties, we can prove the uniformity of the family of functions r_ψ.

▶ **Theorem 27.** *If f is in $\mathsf{Hom}(I,J)$ and ψ is in $\mathbb{F}(J)$, then $r_\psi(f \times \mathrm{id}) = (f \times \mathrm{id})r_{\psi f}$*

This is proved by induction on the number of element of I (the result being clear if I is empty).

A particular case is $r_J(f \times \mathrm{id}) = (f \times \mathrm{id})r_{\partial_J f}$. Note that, in general, $\partial_J f$ is not ∂_I.

A direct consequence of the previous theorem is the following.

▶ **Corollary 28.** *The singular cubical set associated to a topological space has a composition structure.*