Lecture Notes on Cubical sets

March 13, 2015

Introduction

These are some lecture notes for a course presenting the cubical set model of type theory, first in Copenhagen, December 2014, and then in Paris, February 2015.

We describe a particular presheaf model of type theory. This description can also be seen as an operational semantics of a purely syntactical type system. It involves a nominal extension of λ-calculus.

We use a generalization of the Kan composition operation which is represented by a nominal operation. Any element represents a hypercube, the direction being represented by the symbols this element depends on. One can define the notion of homotopy between two cubes (it is a cube with one new dimension connecting the two given cubes). The operation we introduce consists in changing some faces along given homotopies. (Kan composition is the special case where we replace all faces along given homotopies.) One new feature, compared with usual λ-calculus or first-order logic, is how this operation interacts with substitution.

1 Cubical sets

1.1 Definitions

A de Morgan algebra is a bounded distributive lattice $A$, with a top element 1 and a bottom element 0 and with an operation $1 - i$ satisfying

\[ 1 - 0 = 1 \quad 1 - 1 = 0 \quad 1 - (i \lor j) = (1 - i) \land (1 - j) \quad 1 - (i \land j) = (1 - i) \lor (1 - j) \]

This notion differs from the one of Boolean algebra by requiring neither $1 = i \lor (1 - i)$ nor $0 = i \land (1 - i)$.

A prime example of a de Morgan algebra, which is not a Boolean algebra, is the interval $[0,1]$ with $\max(i,j)$, $\min(i,j)$ operations.

We assume a given (discrete) set of symbols/names/directions, not containing 0, 1. We let $I,J,K,\ldots$ denote finite sets of such symbols. Let $C$ be the following category. The objects are finite sets $I,J,K,\ldots$. A morphism $I \to J$ is a map $I \to dM(J)$, where $J$ is the free de Morgan algebra on $J$. We think of $f$ as a substitution and may write $if$ the element $f(i)$ in $dM(J)$. If $f : I \to J$ and $g : J \to K$ we write $fg : I \to K$ the composition of $f$ and $g$. We write $1_I : I \to I$ the identity map. A cubical set is a presheaf on $C^{\text{opp}}$, i.e. a functor $C \to \text{Set}$.

A cubical set $X$ is thus given by a family of sets $X(I)$ together with a restriction map

\[ X(I) \to X(J) \]

\[ u \mapsto uf \]

such that $u1_I = u$ and $(uf)g = u(fg)$. (We write $uf$ for what is usually written $X(f)(u)$, since we want to think about this operation as a substitution; the elements of $X(I)$ for $I = i_1,\ldots,i_n$ are thought of as elements $u = u(i_1,\ldots,i_n)$ depending on $i_1,\ldots,i_n$ and the restriction $uf$ as a substitution operation.) For instance an element $u = u(i,j)$ in $X(i,j)$ represents a square, and if $(i0) : i,j \to j$ is the map sending $i$ to 0, then $u(i0)$ is the face $u(0,j)$ of this square. If $(i = j) : i,j \to j$ is the map sending $i$ to $j$ then $u(i = j)$ is the diagonal $u(j,j)$. 
We write \( \vdash_I A \) if \( A \) is a presheaf on the slice category \( C^{opp}/I \). If \( I \) is empty, we get back a cubical set. If \( I = \{i\} \) then \( A = A(i) \) represents a “line” connecting the cubical sets \( A(0) \) and \( A(1) \). In general, if \( I = \{i_1, \ldots, i_n\} \), then \( A \) represents a hypercube. Concretely, \( A \) is given by a family of sets \( A_f \) indexed by \( f : I \to J \) together with a family of restriction maps \( u \mapsto ug, Af \to Af g \) for \( g : J \to K \) such that \( u1_f = u \) and \((ug)h = u(gh)\) if \( h : K \to L \). If \( \vdash_I A \) and \( f : I \to J \) we can consider \( \vdash_J Af \) which is defined by \( (Af)g = A(fg) \) for \( g : J \to K \).

We write \( \vdash_I a : A \) to mean that \( a \) is an element in the set \( A1_I \). It then defines a family of elements \( af \) in \( Af \).

1.2 Examples

Any topological space \( X \) defines a cubical set, by taking \( X(I) \) to be the set of continuous maps \([0, 1]^I \to X \).

Any (strict) category defines a cubical set, by taking for points the objects of the category, the lines being given by maps, squares given by commuting squares, and so on.

If \( R \) is any de Morgan algebra, a map \( f : I \to J \) defines canonically a map \( R^J \to R^I \) by composing \( R^{dM(J)} \to R^I \) with the extension map \( R^J \to R^{dM(J)} \). Since \([0, 1] \) is a de Morgan algebra, we can in particular define a functor

\[
C^{opp} \to \text{Top}
\]

\[
I \mapsto [0, 1]^I
\]

This can then be used to define the geometric realization functor sending a cubical set to a topological space; this functor commutes with finite products.

The interval \( I \) is the cubical set defined by \( I(J) = dM(J) \). This defines a functor since any map \( I \to dM(J) \) corresponds exactly to a de Morgan algebra map \( dM(I) \to dM(J) \). We can think of \( I \) as an abstract representation of the unit real interval \([0, 1] \), and we have operations \( i \wedge j, i \vee j, 1 - i \) that are abstract representations of the operations \( \min(i, j), \max(i, j), 1 - i \). An element of \( dM(I) \) is determined by a de Morgan formula \( \psi \) on indeterminates in \( I \) and the restriction map \( \psi \mapsto \psi f \) is a substitution.

2 Remarks on the base category

We say that a map \( f : I \to J \) is strict if \( if \) is neither 0 nor 1 for all \( i \) in \( I \). One key remark is the following.

Lemma 2.1 If \( f : I \to J \) is strict and \( \psi \) in \( dM(I) \) such that \( \psi f = b \) (where \( b \) is 0 or 1) then already \( \psi = b \).

(This does not hold if we work with Boolean algebra instead of de Morgan algebra. For instance the map \((i = j) : \{i, j\} \to \{j\} \) is strict and \((i \wedge (1 - j))(i = j) = 0 \) in a Boolean algebra, but \( i \wedge (1 - j) \) is neither 0 nor 1.)

A face map \( \alpha : I \to I_\alpha \) is a map such that \( i\alpha \) is either 0, 1 or \( i \) for all \( i \) in \( I \). We write \( I_\alpha \) the subset of element \( i \) such that \( i\alpha = i \), and \( \text{dom}(\alpha) = I - I_\alpha \) is the domain of \( \alpha \). If \( i_\alpha : I_\alpha \to I \) is the inclusion, we have \( i_\alpha \alpha = 1 \) and hence any face map \( \alpha \) is epi. If \( f : I \to J \) we write \( f \preceq \alpha \) to mean that there exists a map \( f' \) (uniquely determined) such that \( f = \alpha f' \). This means that \( if = i\alpha f \) for all \( i \) in the domain of \( \alpha \). This defines a poset structure on the set of face maps \( \alpha : I \to I_\alpha \) and this poset is a partial meet-semilattice: if \( \alpha \) and \( \beta \) are compatible then they have a meet \( \gamma = \alpha \wedge \beta \) with \( I_\gamma = I_\alpha \cap I_\beta \).

Corollary 2.2 If \( fg \preceq \alpha \) and \( g \) is strict then \( f \preceq \alpha \).

Proof. For any \( i \) in the domain of \( \alpha \) we have \( i\alpha = if g \) and so \( i\alpha = if \) since \( i\alpha = 0 \) or 1 and by Lemma 2.1.

Any map \( f : I \to J \) can be written uniquely as the composition \( f = \alpha h \) of a face map \( \alpha : I \to I_\alpha \) and a map \( h : I_\alpha \to J \) which is strict.
Lemma 2.3 If we have \( \alpha f = \beta g \) with \( f : I_\alpha \to J \) and \( g : I_\beta \to J \) then \( \alpha \) and \( \beta \) are compatible. If \( \gamma \) is the meet of \( \alpha \) and \( \beta \), then there exists a unique \( h : I_\gamma \to J \) such that \( \alpha f = \gamma h = \beta g \). We write \( \alpha_1 \gamma = \beta \gamma_1 \) then \( \alpha_1 f = h = \beta g \).

3 Systems

We define \( S(I) \) to be the set of downward closed subset of face operations on \( I \). An element of \( S(I) \) is determined uniquely by the set of its maximal element, a set \( L \) of incomparable face operations on \( I \). We write \( \alpha \leq L \) to mean that \( \alpha \leq \beta \) for some \( \beta \) in \( L \). If \( L \) is in \( S(I) \) and \( g : I \to J \) we write \( f \leq I \) to mean \( f \leq \alpha \) for some \( \alpha \leq L \). We define then \( \beta \leq L f \) for \( \beta : J \to J_\beta \) and \( f : I \to J \) to mean \( f \beta \leq L \). This define a new downward closed subset of face operations of \( J \). Thus \( S \) can also be seen as a cubical set, since we have \( L1 L = L \) and \( L f g = L(f g) \).

If \( \vdash I A \) and \( L \) in \( S(I) \) a \( L \)-system for \( A \) is given by a family \( a_\alpha \) in \( A \alpha \) which is compatible: if \( \alpha_1 = \beta \gamma_1 \) then \( a_\alpha \alpha_1 = a_\beta \gamma_1 \). We think of such a system as a system of equations \( u \alpha = a_\alpha \) for \( \alpha \) in \( I \). Notice that any element \( v \) in \( \Pi I A \) defines a compatible system \( a_\alpha = \alpha v \). This implies that if \( \alpha f = \beta g \) then we have \( a_\alpha f = a_\beta g \).

If \( f \leq L \) we can define \( a f \) in \( A f \) without ambiguity: if we have both \( f = \alpha g \) and \( f = \beta h \) then \( \alpha \) and \( \beta \) are compatible by Lemma 2.3. A \( L \)-system can also be seen as a compatible family \( a f \) for \( f \leq L \).

If \( f : I \to J \) and we have a \( L \)-system \( (a h) \), \( h \leq L \) we define a \( L f \) system \( b_\beta \) by taking \( b_\beta = a_\beta f \) and more generally \( b_\beta = a g_\beta \).

We adopt the following notations for systems. If for instance \( I = i, j \) and \( \vdash I A \) and \( L \) is determined by \( (i 0) \) and \( (j 1) \), a \( L \)-system \( \vec{a} \) will be determined by an array \( (i 0) \rightarrow u, (j 1) \rightarrow v, \) with \( u \) in \( A(i 0) \) and \( v \) in \( A(j 1) \). If \( f : I \to k, l, j \) is defined by \( f(i) = k \land l, f(j) = j \) then \( L f \) is the system \( (i 0), (j 1) \) and \( \vec{a} f \) is the system \( (i 0) \rightarrow u, (i 0) \rightarrow u, (j 1) \rightarrow v \). If \( g : I \to j \) is defined by \( g(i) = j \) then \( L g \) is empty and \( \vec{a} g \) is the empty system. Notice that \( \vec{a} g(i)(j 1) \) is the system \( (i) \rightarrow u(j 1) \) which is equal to \( (i) \rightarrow v(i 0) \).

For motivations why we introduce such a notion of system, see Appendix 1.

4 Operational semantics

We limit ourselves first to the description of the system without universes. (We describe later the operational semantics for univalence and composition in te universe.) The point is to explain how we can justify function extensionality without using function extensionality at the metalevel.

The syntax for the terms is

\[
\begin{align*}
  t, p, A, E, F ::=: & \quad x \mid t t \mid \lambda x. t \mid \text{Id} A t t \mid \Pi A F \mid (i) t \mid t[p] \mid \text{comp}'(A) \mid t \varphi \\
\end{align*}
\]

where \( \varphi \) represents an element in the free de Morgan algebra on the symbols. In this syntax, \((i) t\) represents the path abstraction operation, and binds the symbol \( i \). We use the vector notation \( \vec{t} \) to represent a system of terms. For instance \( \vec{t} \) may be of the form \((j 0) \rightarrow t, (j 1) \rightarrow u \) or of the form \((j 0) \rightarrow t, (k 0) \rightarrow u, (j 1)(k 1) \rightarrow v \).

The composition operation \( t[p] \) is a new kind of nominal operation. Intuitively, it consists in replacing the face \( t a \), which is equal to \( p a 0 \), by the face \( p a 1 \). The special character of this operation is reflected by the way substitution interacts with it. We have for instance

\[
(a|(j 0) \rightarrow u, (j 1) \rightarrow v)f = (a f|(k 0) \rightarrow u, (l 0) \rightarrow u, (k 1)(j 1) \rightarrow v)
\]

if \( j f = k \land l \). We also have

\[
(a|(j 0) \rightarrow p, (j 1) \rightarrow q)(j 0) = p 1 \quad (a|(j 0) \rightarrow p, (j 1) \rightarrow q)(j 1) = q 1
\]

This operation also satisfies a regularity condition. We have \( a = a[p] \) if all \( p_\alpha \) are constant path, and \( a[p] = a[q] \) if \( q \) is obtained from \( p \) by removing some \( p_\alpha \) that are constant path. This expresses a strict identity element law for this composition.
We define \( p^* = \langle i \rangle p(1-i) \) and 
\[
\overline{p}|a = a|p^2
\]

The operation \( \text{comp}^i(A) \) binds the symbol \( i \). Its intended type is \( A(i0) \to A(i1) \). The regularity condition is that \( \text{comp}^i(A, a_0) = a_0 \) if \( A \) is independent of \( i \). We may write \( \text{comp}^i(A, a_0) \) instead of \( \text{comp}^i(A) a_0 \).

We define \( \text{fill}^i(A, a_0) = \text{comp}^i(A(i\land j), a_0) \) which satisfies \( \Gamma \vdash_{I,i} \text{fill}^i(A, a_0) : A \) and \( \text{fill}^i(A, a_0)(i0) = a_0 \) and \( \text{fill}^i(A, a_0)(i1) = \text{comp}^i(A, a_0) \). So this element represents a line in direction \( i \) connecting \( a_0 \) and \( \text{comp}^i(A, a_0) \).

We can generalize this operation as follows. We define, given a system \( \Gamma \vdash_{I,i} \alpha : A \alpha \) such that \( a_0\alpha = a\alpha(i0) \) the element 
\[
\text{comp}^i(A, a_0, \overline{\alpha}) = u|\overline{\alpha} : A(i1)
\]
where \( u = \text{comp}^i(A, a_0) : A(i1) \) and \( u_\alpha = \langle i \rangle \text{comp}^i(A\alpha(i\lor j), a_\alpha) \). This element satisfies
\[
\text{comp}^i(A, a_0, \overline{\alpha})\alpha = a_\alpha(i1)
\]

We can then define
\[
\text{fill}^i(A, a_0, \overline{\alpha}) = \text{comp}^i(A(i\land j), a_0, \overline{\alpha}(i\land j)) : A
\]
which satisfies
\[
\text{fill}^i(A, a_0, \overline{\alpha})\alpha = a_\alpha \quad \text{fill}^i(A, a_0, \overline{\alpha})(i0) = a_0
\]

We have the usual \( \beta \)-reduction rule
\[
(\lambda x.t) u = t(x = u)
\]

We write \( (x : A) \to B \) for \( \Pi A (\lambda x. B) \).

If \( f : I \to dM(J) \) we can define the operation \( i \mapsto tf \) on terms. (Notice that this is a defined operation on terms; we don’t have an explicit term constructor for substitution.) We have
\[
((\langle i \rangle t) f = (\langle j \rangle t) g)
\]
where \( g : I, i \to dM(J, j) \) extends \( f \) by \( g(i) = j \) not in \( J \). We also have
\[
(\lambda x.t) f = \lambda x.t f \quad x f = x \quad (t u) f = t f u f \quad (t \varphi) f = t f \varphi f \quad (\Pi A F) f = \Pi A f F f
\]

We can then state the path reduction law
\[
(\langle i \rangle t) \varphi = t(i = \varphi)
\]

A canonical object of type \( \text{id} A a b \) is of the form \( \langle i \rangle t \) with \( t(i0) = a \) and \( t(i1) = b \). If \( w \) is of type \( \text{id} A a b \) then \( w\varphi \) is of type \( A \) and \( w0 = a \) and \( w1 = b \).

If \( L \) is a system for \( I \) we write \( p : \text{id}^L A u v \) or \( p : u \sim_{L} v \) to express that we have \( p0 = u \) and \( p1 = v \) and each \( p_\alpha \) is constant for \( \alpha \leq L \).

We define \( a \uparrow \overline{p} \) to be \( \langle i \rangle a|\overline{q} \) where \( q_\alpha = \langle j \rangle p_\alpha(i \land j) \). Using regularity we have that \( a \uparrow \overline{p} \) defines a line \( l : a \sim (a|\overline{p}) \) such that \( l_\alpha = p_\alpha \). We also define \( \overline{p} \uparrow a = a \uparrow \overline{p^2} \).

The main new computation rules are for the composition of a product type and the composition of an identity type.

### 4.1 Dependent products

For dependent product types, we have
\[
(w|\overline{p}) a = (w a)|\overline{q}
\]
where \( q_\alpha = \langle i \rangle p_\alpha(i, a\alpha) \).

This provides a short and effective justification of the fact that homotopy types are closed by exponentiation. The reader can compare this argument with the argument in [6, 4].

We also have
\[
\text{comp}^i(\Pi A F, w_0) u_1 = \text{comp}^i(F u, w_0 u(i0))
\]
where \( u = \text{fill}^i-\langle i \rangle(A, u_1) \).
4.2 Identity types

For identity types, we have

\[(p|q) \varphi = p \varphi|p\]

where \(r_\alpha = \langle i \rangle q_\alpha (i, \varphi)\).

We also have

\[\text{comp}'(\text{id} A a b, p_0) = \langle j \rangle \text{comp}'(A, \bar{u}, p_0 j)\]

\(\bar{u}\) is the system \((j0) \mapsto a, (j1) \mapsto b\).

We can interpret

\[\text{id} A a_0 a_1 \rightarrow B(a_0) \rightarrow B(a_1)\]

Indeed if \(p\) is of type \(\text{id} A a_0 a_1\) and \(b_0 : B(a_0)\) then \(\text{comp}'(B(p i), b_0)\) is of type \(B(a_1)\).

Using the operation \(i \land j\) on symbols we can interpret the fact that \((x : A, \text{id} A a x)\) is contractible. Indeed, if \(x, p\) is an element of this type then

\[q = \langle i \rangle (p(i), (j)p(i \land j))\]

is a path such that \(q(0) = (a, (j)a)\) and \(q(1) = (x, (j)p(j)) = (x, p)\). If \(x = a\) and \(p = \langle i \rangle a\) (which interprets reflexivity) we get \(q = \langle i \rangle (a, (j)a)\) which is a constant path.

We can then interpret the usual \(J\) elimination rule. Because of the regularity condition, the computation rule for \(J\) is interpreted as a judgemental equality.

5 Typing rules

We have judgement of the forms \(\Gamma \vdash_I, \Gamma \vdash_{1I} A\) and \(\Gamma \vdash_I t : A\) relativized at a “level” (finite set of symbols) \(I\). The rules are the usual rules of type theory at all levels \(I\), and the restriction rule

\[
\frac{\Gamma \vdash_I t : A}{\Gamma \vdash_{1I} tf : A_{}\Gamma}
\]

is admissible.

The new rules are then the following.

\[
\frac{\Gamma \vdash_I a : A \quad \Gamma \alpha \vdash_{1I} p_\alpha : \text{id} A \alpha \alpha \ u_\alpha}{\Gamma \vdash_I a[p] : A}
\]

with a computation rule \((a[p]\alpha) = u_\alpha\). We also have

\[
\frac{\Gamma \vdash_I A \quad \Gamma_0 \vdash_{1I} a_0 : A \quad \Gamma \vdash_I a_1 : A}{\Gamma \vdash_I \text{id} A a_0 a_1}
\]

\[
\frac{\Gamma_0 \vdash_I A \quad \Gamma_0 \vdash_{1I} t : A}{\Gamma \vdash_I \langle i \rangle t : \text{id} A t(i0) t(i1)}
\]

In particular, we get the reflexivity proof of \(a : A\) as the constant path \(\langle i \rangle a\)

\[
\frac{\Gamma \vdash_I \text{id} A a_0 a_1}{\Gamma \vdash_I \langle i \rangle t : \text{id} A a_0 a_1}
\]

\[
\frac{\Gamma \vdash_I \langle i \rangle t : \text{id} A a_0 a_1}{\Gamma \vdash_I \langle i \rangle t 0 = a_0 : A}
\]

\[
\frac{\Gamma \vdash_I \langle i \rangle t 1 = a_1 : A}{\Gamma \vdash_I \text{comp}'(A) : A(i0) \rightarrow A(i1)}
\]

We can justify function extensionality by the defined constant

\[
\frac{\Gamma \vdash_I t : (x : A) \rightarrow B \quad \Gamma \vdash_I u : (x : A) \rightarrow B \quad \Gamma \vdash_I p : (x : A) \rightarrow \text{id} B \ (t x) (u x)}{\Gamma \vdash_I \text{ext} t u p : \text{id} \ ((x : A) \rightarrow B) \ t u}
\]

where \(\text{ext} \ t \ u \ p = \langle i \rangle \lambda x. p \ x \ i\).
6 Equivalence

We say that \( \vdash \sigma : T \rightarrow A \) is an equivalence if, given a \( L \)-system \( \vec{t} \) in \( T \), and \( a : A \) such that \( a \alpha = \sigma \alpha t_\alpha \), we can find \( t \) in \( T \) such that \( t \alpha = t_\alpha \) and a path in \( A \) showing \( a \sim_L \sigma \ t \). Furthermore this operation transforming \( a \) and \( \vec{t} \) to \( u \) has to be uniform.

6.1 Isomorphisms

We introduce a type \( \text{Iso}(A,B) \) of isomorphisms between \( A \) and \( B \). An element of \( \text{Iso}(A,B) \) is a tuple \((\sigma, \delta, \eta, \epsilon)\) where \( \sigma : A \rightarrow B \) and \( \delta : B \rightarrow A \) and \( \eta : \sigma \delta a \rightarrow a \) and \( \epsilon : \delta \sigma t \rightarrow t \). If \( u : \text{Iso}(A,B) \) we write \( u^+ : A \rightarrow B \) and \( u^- : B \rightarrow A \) the correspondig functions.

6.2 Graduate Lemma

Lemma 6.1 If \( u : \text{Iso}(A,B) \) then \( u^+ : A \rightarrow B \) is an equivalence.

This corresponds to the "graduate lemma", and it has a rather direct proof.

If \( u \) is the identity isomorphism, then the \( L \)-path produced by the corresponding equivalence is a constant path.

Lemma 6.2 If \( E : T \rightarrow _A \) then \( \text{comp'E} \) is an equivalence.

7 Representation of cubical sets

We define an \( I \)-element to be a tuple \( u = (u_\alpha) \) indexed by all face operations \( \alpha : I \rightarrow I_\alpha \). If all elements of a set \( X \) are \( I \)-element, we say that \( X \) is a \( I \)-set. If \( u \) is a \( I \)-element and \( \alpha : I \rightarrow I_\alpha \) is a face of \( I \) we can define an \( I_\alpha \) element \( u_\alpha \) by taking \( u_{\alpha\beta} = u_{\alpha\beta} \).

If \( a \) is an \( I \)-element and \( \vec{v} \) a system of \( I_\alpha \) element \( v(\alpha) \) which is compatible, i.e. \( v(\alpha) \beta_1 = v(\beta) \alpha_1 \) whenever \( \alpha \beta_1 = \beta \alpha_1 \), then we define an \( I \)-element \((\vec{v}, a)\) by taking \((\vec{v}, a)_\alpha = v(\beta)_\alpha \) if \( \alpha = \beta \alpha_1 \) and \((\vec{v}, a)_\alpha = a_\alpha \) otherwise. This operation satisfies \((\vec{v}, a) = a \) if \( v(\alpha) = a \alpha \).

We consider only presheaves \( A \) on \( C^{opp} \) such that

1. all elements of \( A(I) \) are \( I \)-element
2. for any \( u \) in \( A(I) \) we have \( u_{\alpha\beta} = u_{\alpha\beta} \)

It can be checked that all type-forming operations produce objects of this form. For instance if \( F \) and \( G \) are any presheaves on \( C \) then \( C^{opp}(I) \) is a set of sequences \( \lambda_f \) in \( F(J) \rightarrow G(J) \) for \( f : I \rightarrow J \), satisfying the condition \( (\lambda_f u)g = \lambda_{fg} u_g \), and hence \( \lambda \) can be written as a tuple \((\lambda_\alpha)\) with \( \lambda_\alpha = (\lambda_{\alpha\beta}) \) indexed by strict map \( \alpha : I \rightarrow J \). This follows from the fact that a map can be uniquely written as the composition of a projection and a strict map.

If we have \( \vdash_I A \) and \( L \) is an element of \( S(I) \) and we have \( \vdash_I \sigma \alpha : T_\alpha \rightarrow A_\alpha \) we define a new type \( \vdash_I B = \bar{\sigma}[A] \). For \( f : I \rightarrow J \), an element of \( BF \) is a \( J \)-element \((\vec{f}, u)\) where \( u \) is in \( AF \) and we have \( \sigma_\beta t_\beta = u_\beta \) for \( \beta \) in \( LF \).

If \( L \) is empty then \( \bar{\sigma}[A] = A \). If \( L \) has one element () then \( \bar{\sigma}[A] = T_1 \).

If each \( \sigma_\alpha \) is the identity map then \( A = \bar{\sigma}[A] \) as a cubical set.

This basic operation will be used to define glueing (which transforms equivalence to equality) and the composition operation in the universe. In each case, we will get the same underlying type if all maps are identities. In the case of glueing however, the Kan composition operations does not need to stay the same, while it will be the same for composition, which ensures regularity for composition in the universe.

If we have \( B = \bar{\sigma}[A] \) then there is a canonical map \( \delta : B \rightarrow A \). For instance, if \( I = i \) and we have \( \sigma_0 : T_0 \rightarrow A_0 \) then \( B \) is the set of sequences \((t_0, a_{i1}, a_1)\) such that \( a_1 \) is in \( A_1(\sigma_0 t_0, a_{i1}) \) and we define \( \delta(t_0, a_{i1}, a_1) = (\sigma_0 t_0, a_{i1}, a_1) \).
8 Glueing operation

We introduce an operation on the universe, which consists in changing some faces of a given cube along isomorphisms. As a particular case, we can transform one given isomorphism to an equality between two types.

Lemma 8.1 If we have \( \vdash_{t, \alpha} \sigma : T \to A \) then we have a path \( \sigma(i_1) \) \((\text{comp}^i(T, t_0)) \to \text{comp}^i(A, \sigma(i_0) \ t_0)\).

*Proof.* We define a system \( \vec{w} \) as \((j \in ) \rightarrow \sigma(\text{fill}^j(T, t_0)), (j \in ) \rightarrow \text{fill}^j(A, \sigma(i_0) \ t_0)\) and \(p = (j)\text{comp}^i(A, \vec{w}, a_0)\) is a path \( u(i_1) \to v(i_1)\).

Lemma 8.2 If we have \( \vdash_{t, \alpha} \sigma : T \to A \) then for any system of paths \( \vec{p} \) compatible with \( t : T \) we have \( \sigma(t[i]) \sim_L \sigma(t[i]) \) where \( q_{\alpha} = \langle i \rangle \sigma_{\alpha},(i)\).

In order to interpret univalence we explain how to transform an equivalence to an equality.

More generally, given a \( L \)-system \( \vec{T} \) in \( U \) and a type \( A \) together with a compatible system of isomorphisms \( u_{\alpha} : \text{iso}(A_{\alpha T_{\alpha}}) \) we define a new type

\[
B = A[\vec{u}]
\]

As a cubical set \( B \) is \( \vec{\sigma}|A \) where \( \sigma_{\alpha} = u_{\alpha}^{-1} \). An element of this type is of the form \((\vec{r}, a)\) with \( t_{\alpha} : T_{\alpha} \) and \( a : A \) such that \( aa = \sigma_{\alpha}t_{\alpha} \). If \( L = () \), we only have one equivalence \( \sigma() : T() \to A \) and \( B = T() \). If \( L \) is empty we have \( B = A \). If \( f : I \to J \) we define

\[
Bf = Af[\vec{u}f]
\]

If we have one equivalence \( u : \text{iso}(A, T) \), then introducing a fresh symbol \( i \), we have \( A = A(i0) \) and \( B = A(i0) \to u \). This type \( B \) will be such that \( B(i0) = T \) and \( B(i1) = A(i1) = A \). So we get in this way an operation transforming an equivalence to an equality.

We have a map \( \delta : B \to A \) such that \( \delta(\vec{r}, a) = a \) and \( \delta a = \sigma_{\alpha} t_{\alpha} \) for \( \alpha \) in \( L \).

Proposition 8.3 The type \( B = A[\vec{u}] \) has composition operations.

*Proof.* We have to define \( v_0[\vec{q}] \) for \( v_0 : B \) and a \( J \)-system \( \vec{q} \). Using the map \( \delta \), we define a \( J \)-system \( \vec{p} \) in \( A \) by \( p_{\alpha} = (i)\delta a_{\alpha}(i) \) and \( a_0 = \delta v_0 \).

We consider \( a_1 = a_0[\vec{p}] \). For \( \beta \) in \( J \) we have \( a_1 \beta = p_\beta(1) \).

The goal is to build \( v_1 = v_0[\vec{q}] \) in \( B \).

We should have, for each \( \alpha \) in \( L \), \( t_{\alpha} = v_1 = v_1 = v_0 = v_0[\vec{q}] \). On the other hand, we have \( a_1 \alpha = a_0 \alpha[\vec{p}] \) in \( A_{\alpha} \).

Using Lemma 8.2, we have that \( a_1 \alpha \sim_{J_{\alpha}} \sigma_{\alpha} t_{\alpha} \) and so, we have a line \( r_{\alpha} : a_1 \alpha \to \sigma_{\alpha} t_{\alpha} \) constant on \( J_{\alpha} \). We define \( v_1 = (\vec{r}, a_1[\vec{r}] \).

Notice that the argument uses only that we have compatible functions \( T_{\alpha} \to A_{\alpha} \) and not that these functions are equivalences.

Proposition 8.4 The type \( B = A[\vec{u}] \) has transport functions.

*Proof.* We have to define \( \text{comp}^i(B, v_0) : B(i1) \) for \( v_0 : B(i0) \). We define \( a_0 = \delta(i0) \) \( v_0 \) and \( a_1 = \text{comp}^i(A, a_0) \).

Let \( L' \) be the subset of \( \gamma \) in \( L \) not mentioning \( i \) and \( L'' \) subset of \( \gamma \) such that \( \gamma(i1) \) in \( L \). Since the element in \( L \) are incomparable, \( L' \) and \( L'' \) are disjoint but it may be that some element in \( L' \) is \( < \) some element in \( L'' \). We have that \( L(i1) \) is the union of \( L'' \) and of \( L'_i \) subset of element in \( L' \) not \( < L'' \).

We want to write \( v_1 = (\vec{r}, u_1) \) for some \( u_1 \) in \( A(i1) \) which should be obtained from \( a_1 \) by modifying some faces in \( L' \) and \( L'' \). What are the constraints on this element \( u_1 \)?

For \( \gamma \) in \( L' \), we should have \( u_{\alpha1} = \sigma_{\alpha}(i1) t_{\alpha} \) where \( t_{\gamma} = \text{comp}^\gamma(T_{\gamma}, v_{\gamma}, v_{\gamma}) \) is of type \( T_{\gamma}(i1) \).

For \( \gamma \) in \( L'' \), then \( u_{\alpha1} \gamma \) should be of the form \( \sigma_{\gamma(i1)} t_{\gamma} \) for some \( r_{\gamma} \) in \( T_{\gamma}(i1) \).
We first deal with the constraints for $\gamma$ in $L'$. We have

$$a_0 \gamma = \sigma_\gamma(i0)v_0 \gamma \quad a_1 \gamma = \text{comp}^i(A\gamma, a_0 \gamma) \quad t_\gamma = \text{comp}^i(T\gamma, v_0 \gamma)$$

We can build a line $w_0 : \sigma_\gamma(i1)t_\gamma \rightarrow a_1 \gamma$ using Lemma 8.1. (There is no reason for this line to be constant even if $\sigma_\gamma$ is the identity map. This is why we need another argument for composition in the universe.)

We define $a'_1 = \bar{w}a_1$. We have $a'_1 \gamma = \sigma_\gamma(i1)t_\gamma$ for all $\gamma$ in $L'$.

The second step deals with the element $\gamma$ in $L''$. For such an element $\gamma$ we have $\alpha = \gamma(i1)$ in $L$. For $\beta$ in $L'$ we have $\gamma\beta \subseteq L'$ and we also have $a'_1 \gamma \beta = \sigma_\alpha \beta t_\beta$ for some $t_\beta$ in $T\alpha \beta$. Since $\sigma_\alpha$ is an equivalence we can build $t_\gamma$ in $T\gamma(i1) = T\alpha$ and a line $s_\gamma : \sigma_\alpha t_\gamma \rightarrow a'_1 \gamma$ which is a $L'\gamma$-path. We change then $a'_1$ to $u_1 = \bar{s}a'_1$.

By regularity, we have $u_1 \delta = a'_1 \delta$ for $\delta$ in $L'$, so we did not modify the $L'$ faces of $a'_1$.

The element $u_1 = \bar{s}a'_1$ satisfies $u_1 \gamma = \sigma_\gamma t_\gamma$ for $\gamma$ in $L'$ and $u_1 \gamma = \sigma_\gamma(i1)t_\gamma$ for $\gamma$ in $L''$. Hence, we can define a corresponding element $v_1 = (\vec{v}, u_1)$ in $B(i1)$ with $r_\gamma = t_\gamma$ in $T\gamma(i1)$ for $\gamma$ in $L'_1$ and $r_\gamma = t_\gamma$ in $T\gamma(i1)$ for $\gamma$ in $L''$.

\[\Box\]

The corresponding typing rules are

\[
\begin{array}{c}
\Gamma \vdash I \quad \Gamma \vdash I_\alpha u_\alpha : \text{ls}(A\alpha, T_\alpha) \\
\hline
\Gamma \vdash I A|\vec{u}
\end{array}
\]

\[
\begin{array}{c}
\Gamma \vdash I a : A \\
\Gamma \vdash I_\alpha u_\alpha : \text{ls}(A\alpha, T_\alpha) \\
\Gamma \vdash I_\alpha u_\alpha t_\alpha = a\alpha : A\alpha
\end{array}
\]

\[
\begin{array}{c}
\Gamma \vdash I (\bar{v}, a) : A|\vec{u}
\end{array}
\]

We also have

\[
\begin{array}{c}
\Gamma \vdash I b : A|\vec{u}
\end{array}
\]

\[
\begin{array}{c}
\Gamma \vdash I (\bar{v}, b) : A
\end{array}
\]

where $v_\alpha = u_\alpha b\alpha$.

### 9 Composition in the universe

It is almost the same operation as for gluing. We assume given a type $A$ and paths $E_\alpha : A\alpha \rightarrow T_\alpha$. and we explain how to build $B = A|\vec{E}$.

We have a corresponding equivalence $\sigma_\alpha : T_\alpha \rightarrow A\alpha$ using Lemma 6.1. The type $B$ is defined as $A|\vec{E}$ as a cubical set. As for gluing, we define an element of $B$ to be of the form $(\vec{t}, a)$ with $a$ in $A$ and $t_\alpha$ in $T_\alpha$ such that $\sigma_\alpha t_\alpha = a\alpha$. The difference between the previous case of gluing is how we define the composition for $A|\vec{E}$.

**Lemma 9.1** We assume given $A,T,E$ with $E(i0) = T$ and $E(i1) = A$ and we define $\sigma : T \rightarrow A$ by $\sigma t = \text{comp}^i(E,t)$. Given $t_0$ in $T(i0)$ we have a path $\text{comp}^i(A,\sigma(i0)t_0) \rightarrow \sigma(i1)\text{comp}^i(T, t_0)$. Furthermore this path is constant if $E$ is independent of $j$.

**Proof.** We define $e_0 = \text{fill}^i(E(i0), t_0)$ and $t_1 = \text{comp}^i(T, t_0)$. If $e_1 = \text{comp}^i(E, e_0)$ we have $e_1(j0) = t_1$ and $e_1(j1) = \text{comp}^i(A, \sigma(i0)t_0)$.

We define next $e'_1 = \text{fill}^i(E(i1), t_1)$ so that $e'_1(j0) = t_1$ and $e'_1(j1) = \sigma(i1)t_1$. We can then consider

$$\langle k \rangle (t_1(k0) \rightarrow (j)e_1, \ (k1) \rightarrow (j)e'_1)$$

which is a path $\text{comp}^i(A, \sigma(i0)t_0) \rightarrow \sigma(i1)t_1$. If $E$ is independent of $j$ then so are $e_1$ and $e'_1$ and this is the constant path $\langle k \rangle t_1$.

\[\Box\]

**Lemma 9.2** We assume given $A,T,E$ with $E(i0) = T$ and $E(i1) = A$ and we define $\sigma : T \rightarrow A$ by $\sigma t = \text{comp}^i(E,t)$. We have for any $L$-system of paths $\vec{p}$ and $t_0$ in $T$ that $t_0|\vec{p} \sim_L \sigma t_0|\vec{p}$. Furthermore this path is constant if $E$ is independent of $j$.
Proof. We define in \( E \alpha q_\alpha = \langle i \rangle \text{fill}^I(E \alpha, p_\alpha(i)) \) and \( e_0 = \text{fill}^I(E(i0), t_0) \). and \( e_1 = e_0[\vec{q}] \) and \( t_1 = t_0[\vec{p}] \). We have \( e_1(j0) = t_1 \) and \( e_1(j1) = \sigma t_0[\vec{\sigma p}] \).

We define \( e_1' = \text{fill}^I(E, t_1) \) so that \( e_1'(j0) = t_1 \) and \( e_1'(1) = \sigma t_1 \). We can then consider
\[
\langle k \rangle (t_1)(k0) \mapsto (j)e_1, \quad (k1) \mapsto (j)e_1'
\]
which is a path \( \text{comp}^I(A, \sigma t_0) \to \sigma t_1 \). If \( E \) is independent of \( j \) then so are \( e_1 \) and \( e_1' \) and this is the constant path \( \langle k \rangle t_1 \).

The operation of composition and transport are almost the same as for glueing. The difference is that we use Lemmas 6.1, 9.1 and 9.2 instead of Lemmas 6.2, 8.1 and 8.2.

The corresponding typing rules are
\[
\frac{\Gamma \vdash I A \quad \Gamma' \vdash L_a E_\alpha : \text{ld} U A_a T_\alpha}{\Gamma \vdash I A[l']}
\]

\[
\frac{\Gamma \vdash I a : A \quad \Gamma' \vdash L_a E_\alpha : \text{ld} U A_a T_\alpha \quad \Gamma' \vdash L_a \text{comp}^I(E_\alpha^i, t_\alpha) = a_\alpha : A_\alpha}{\Gamma \vdash I (l', a) : A[l']}
\]

10 Circle

We describe \( S^1 \) as a higher inductive type.

We need to define a set \( S^1(I) \) for each finite set of symbols \( I \). An element of this set is
1. either \text{base}
2. or \text{loop} \( \varphi \) where \( \varphi \) is an element of \( \text{dM}(I) \) different from 0, 1
3. or of the form \( u_0[\vec{p}] \) where \( u_0 \) is in \( S^1(I) \) and \( p_\alpha \) is of the form \( \langle i \rangle u_\alpha \) with \( u_\alpha \) in \( S^1(I, i) \), and such that \( u_\alpha(i0) = u_0\alpha \)

Thus the element of \( S^1(I) \) are defined by these generators together with the relation that we have \( \text{comp}^I(u, u_0) = u_0 \) if all \( u_\alpha \) are independent of \( i \).

We define recursively on \( u \) in \( S^1(I) \) at the same time the element \( u f \) in \( S^1(J) \) if \( f : I \to J \). In this way we interpret \( \vdash S^1 \) with \( \vdash \text{base} : S^1 \) and \( \vdash \text{loop} i : S^1 \). For the cubical set \( S^1 \) it is decidable if \( u \in S^1(I) \) is independent or not of some element \( i \) in \( I \).

Given \( S^1 \vdash F \) it is also possible to define a section \( \vdash s : (\Pi x : S^1)F(x) \) if we give \( \vdash b : F \text{ base} \) and \( \vdash l : F (\text{loop} i) \). Furthermore, we have \( \vdash s \text{ base} = b : F \text{ base} \text{ and } \vdash l (\text{loop} i) = l : F (\text{loop} i) \).

11 Propositional truncation

We describe now the propositional truncation as an element of type \( U \to U \). We define \( U(I) \) to be the set of small \( A \) such that \( \vdash I A \). Concretely \( A \) is a family of small sets \( A f \) with restriction maps \( A f \to A f g, \ u \mapsto u f s \) satistyng \( u_1 = u \) and \( (u f)g = u f g \). Given such a structure \( A \), we have to define a family of sets \( \text{inh}(A) f \). An element of \( \text{inh}(A) f \) is defined inductively as before, it is
1. either \text{inc} \( u \) with \( u \) in the set \( A f \)
2. or \text{squash} \( \varphi \ u_0 \ u_1 \) with \( \varphi \) in \( \text{dM}(J) \) and \( u_0, u_1 \) in \( \text{inh}(A) f \)
3. or of the form \( u_0[\vec{p}] \) where \( u_0 \) is in \( \text{inh}(A) f \) and \( p_\alpha \) is in \( \text{Path}(\text{inh}(A)) f \alpha \)

It is then possible to define \( u g \) in \( \text{inh}(A) f g \) for \( g : J \to K \) by induction on \( u \) in \( \text{inh}(A) f \). We also have \( \text{inh}(A) f g = \text{inh}(A) f g \) and hence we have defined a natural transformation \( \text{inh} : U \to U \).

We show next that \( \text{inh}(A) \) has a transport function if \( A \) has transport function. (This seems to be closely connected to Lemmas 6.2.3 and 6.2.4 in [2].)

We define \( \text{comp}^I(v_0) \) in \( \text{inh}(A) f j 1 \) if \( v_0 \) is in \( \text{inh}(A) f j 0 \). Let \( t r \) be the transport function \( v \mapsto \text{comp}^I(v_0) \). The definition of \( t r(v_0) \) is done by induction on \( v_0 \):
1. in the case where \( v_0 \) is in the form \( \text{inc}(a_0) \) we only need that \( A \) has transport

2. in the case where \( v_0 \) is of the form \( \text{squash} \varphi u_0 u_1 \) then \( tr(v_0) \) is \( \text{squash} \varphi tr(u_0) tr(u_1) \)

3. in the case of \( v_0 \) is of the form \( \text{comp}^j(\vec{u}, u_0) \) where \( j \) not in \( J \) then \( tr(v_0) \) is \( \text{comp}^j(tr(\vec{u}), tr(u_0)) \)

The definition of the suspension operation is similar. The definition of the push-out operation involves new complications for defining the transport function.

12 Fibrant replacement

A slight modification of the previous example gives the fibrant replacement \( \tilde{A} \) of a cubical set which has transport functions. An element of this type will be

1. either \( \text{inc} u \) with \( u \) in the set \( Af \)

2. or of the form \( u_0|\vec{p} \) where \( u_0 \) is in \( \tilde{A}f \) and \( p_\alpha \) is in \( \text{Path}(\tilde{A})f_\alpha \)

Acknowledgement

Many thanks to Georges Gonthier and Thomas Streicher for comments and multiple corrections. In particular, Georges Gonthier noticed a problem with the use of Boolean algebra instead of de Morgan algebra in the first version of this note. Many thanks also to Rasmus Mogelberg and to Pierre-Louis Curien for comments and corrections.

Appendix 1: Motivations for the notion of system

Compared to the original approach by Kan [5] we add new composition operations in order to express how composition interacts with (symbol) substitution.

Let us consider a composition \( u|(i0) \mapsto q, (i1) \mapsto r \)

Intuitively, we replace \( a \) by \( c \) and \( b \) by \( d \), and we get a new line connecting \( c \) and \( d \). We have to consider the degenerate of this composition to be given by

where we replace the constant face \( a \) by \( c \) and the constant face \( b \) by \( d \).
The connection [3] of the same diagram should be

where we replace the point \( a \) by \( c \) and two constant faces \( b \) by \( d \).

If \( f : i \to i, j \) is defined by \( f(i) = i \lor j \) this corresponds to the equality

\[
(u((i0) \mapsto q), (i1) \mapsto r) f = u(i \lor j)((i0)(j0) \mapsto q), (i1) \mapsto r, (j1) \mapsto r
\]

### Appendix 2: Kan cubical sets

#### 12.1 Kan composition operation

A cubical set \( A \) has **Kan composition operations** iff we have a family of operations \( u_0|p \) which takes \( u_0 \) in \( A(I) \) and \( p \) a \( L \)-system for \( \text{Path}(A) \) such that \( p_\alpha 0 = u_0 \alpha \) and produces an element in \( A(I) \). This should satisfy the uniformity condition \( (u_0|p)f = u_0f|pf \), and such that \( u_0|p = q 1 \) if \( p \) is a system of the form \( () \mapsto q \).

#### 12.2 Combinatorial definition of homotopy groups

Given a cubical set \( X \) with composition operation and a point \( a \) in \( X() \) we define \( \pi_1(X, a) \) as follows. The elements are homotopy equivalent classes of path \( a \to a \) and two paths \( p, q : a \to a \) are equivalent iff we can fill a square

We define then the composition of two paths \( p, q : a \to a \) as being the path obtained by the Kan composition

We can write \( p \cdot q = (i)(p i|(i1) \mapsto q) \).

It is then possible to show in a purely combinatorial way that this defines a group. The proof is simpler than in [8] since we have connections. For instance, if \( p : a \to a \) we have
and the following square shows that $p^* = \langle i \rangle (p(1 - i))$ is an inverse of $p$

Since it is clear how to define combinatorial the loop space $\Omega(X, a)$ we get in this way a simple combinatorial definition of $\pi_2(X, a) = \pi_1(\Omega(X, a), 1_a), \ldots$

We can consider a cubical set with Kan composition operations to be a combinatorial representation of homotopy types. Contrary to Kan’s original definition, we can show that this notion of homotopy types is closed under exponential in a constructive meta-framework.

12.3 Reasoning with equality proofs

This cubical description of homotopy types suggest the following “diagrammatic” way to reason about equality proofs. This suggests a direct cubical syntax for representing these proofs, which can be shorter than the proof we obtained using the elimination rule $J$.

Let us show by diagram that $p \cdot q = p \cdot r$ implies $q = r$. We have two filled squares

The cube
shows then that \( p \cdot q \) is equivalent to \( p \cdot r \) if, and only if, \( q \) is equivalent to \( r \).

The proof we get in this way is more direct than the proof obtained using the elimination rule for identity types.

### 12.4 Combinatorial description of \( S^2 \)

A possible combinatorial description of \( S^2 \) as a cubical set is the following. An element of the the set \( S^2(I) \) is

1. either base

2. or loop \( \varphi, \psi \) where \( \varphi \) and \( \psi \) are element of \( dM(I) \) different from 0,1

3. or of the form \( u_0[p] \) with \( u_0 \) is in \( S^2(I) \) and \( p_{\alpha} = (i)u_{\alpha} \) with \( u_{\alpha} \) is a family in \( S^2(I_{\alpha}, i) \) such that \( u_{\alpha}(i0) = u_{00} \alpha \). Furthermore \( i \) should appear free in all \( u_{\alpha} \).

At the same time, we define the substitution operation \( uf \) in \( S^2(J) \) if \( f : I \to J \). For instance we have \( (\text{loop } i \cdot j)(i0) = \text{base} \) and \( (\text{loop } (i \lor j)(i \land j))(i0) = \text{base} \).

It can then be proved in a purely combinatorial way that \( \pi_1(S^2, \text{base}) \) is trivial.

### Appendix 3: General remarks about the model

The first remark is that all paths in \( N \) are constant, as expected.

**Proposition 12.1** \( I \) is the presheaf defined by \( I(J) = dM(J) \) and \( N \) is the constant presheaf \( N(J) = N \). Any natural transformation \( I \to N \) is constant and is determined by the image of \( i \) by the map \( I(\{i\}) \to N \).

The second remark is that one cannot hope to have the right lifting property for monomorphisms against trivial fibrations. If we had this property, we could do the following operation. (I learnt this from Vladimir Voevodsky.)

**Proposition 12.2** For any map \( f : A \to B \) if we have \( a : A \) and \( b : B \) and a path \( f a \to b \) then we can find \( g : A \to B \) such that \( g a = b \) with a path \( f \to g \).

Indeed we can consider the trivial fibration \( (y : B, \text{Id } B (f x) y) \), \( x : A \) and the monomorphism \( 1 \to A \) defined by \( a : A \). We have a map \( (b, q) : 1 \to (y : B, \text{Id } B (f a) y) \). If we had the right lifting property we could find a lifting map \( (x : A) \to (y : B, \text{Id } B (f x) y) \), \( x \mapsto (g x, p x) \) such that \( p x : \text{Id } B (f x) (g x) \) and \( g a = b \) and \( p a = q \).

However, it is not possible to have such a map \( g \) in general as is shown by the following Kripke model over \( 0 \leq 1 \). At time 0 let \( A \) have two distinct points \( a \) and \( a' \) which becomes equal at time 1. Let \( B \) be the groupoid having two connected component \( u \to b \) and \( u' \) at time 0 and only one \( u' = u \to b \) at time 1. We then have a map \( f : A \to B \) taking \( f a = u \) and \( f a' = u' \) and we have a path \( f a \to b \), but there is no map \( g : A \to B \) such that \( g a = b \) with a path \( f \to g \). (Notice that this counter-example holds already in the strict groupoid model: already for this model, we cannot have a constructive model structure.)

13
Appendix 4: New Typing Rules

We collect all new rules of our system.

\[
\Gamma \vdash I a : A \quad \Gamma \vdash p_\alpha : \text{Id} \ A \alpha \ a_\alpha \ u_\alpha \\
\Gamma \vdash I a \vec{p} : A
\]

\[
\Gamma \vdash I A \quad \Gamma \vdash a_0 : A \quad \Gamma \vdash a_1 : A \\
\Gamma \vdash \text{Id} \ A \ a_0 \ a_1 \quad \Gamma \vdash I \ a \ i : \ A \\
\Gamma \vdash \text{Id} \ A \ a_0 \ a_1 \quad \Gamma \vdash I \ t \ 0 = a_0 : A \\
\Gamma \vdash I \ t \ 1 = a_1 : A
\]

\[
\Gamma \vdash I, i \ A
\]

\[
\Gamma \vdash \text{comp}^i(\vec{A}) : A(i0) \to A(i1)
\]

\[
\Gamma \vdash I A \quad \Gamma \vdash u_\alpha : \text{iso}(A\alpha, T\alpha) \\
\Gamma \vdash I A[\vec{u}]
\]

\[
\Gamma \vdash I a : A \quad \Gamma \vdash u_\alpha : \text{iso}(A\alpha, T\alpha) \\
\Gamma \vdash u_\alpha \comp i t = a_\alpha : A
\]

\[
\Gamma \vdash I \vec{t}, a : A[\vec{u}]
\]

\[
\Gamma \vdash I b : A[\vec{u}] \\
\Gamma \vdash I (\vec{t}, b) : A \quad \text{where } v_\alpha = u_\alpha b_\alpha
\]

\[
\Gamma \vdash I A \quad \Gamma \vdash \text{Id} U A \alpha T\alpha \\
\Gamma \vdash I A[\vec{E}]
\]

\[
\Gamma \vdash I a : A \quad \Gamma \vdash \text{Id} U A \alpha T\alpha \\
\Gamma \vdash \text{comp}^i(\vec{E}_\alpha, t_\alpha) = a_\alpha : A
\]

\[
\Gamma \vdash I (\vec{t}, a) : A[\vec{E}]
\]

References


