Lecture Notes on Cubical sets

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 \( \lambda \)-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. One new feature, compared with usual \( \lambda \)-calculus or first-order logic, is how this operation interacts with nominal 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

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

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 \( I \rightarrow J \) is a map \( I \rightarrow dM(J) \), where \( J \) is the free de Morgan algebra on \( J \). We think of \( f \) as a substitution and may write \( uf \) the element \( f(i) \) in \( dM(J) \). If \( f : I \rightarrow J \) and \( g : J \rightarrow K \) we write \( fg : I \rightarrow K \) the composition of \( f \) and \( g \). We write \( 1_I : I \rightarrow I \) the identity map. A cubical set is a presheaf on \( C^{\text{opp}} \), i.e. a functor \( C \rightarrow \text{Set} \).

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

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

\[
u \mapsto uf
\]

such that \( u 1_I = u \) 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 \((i0) : i, j \rightarrow j \) is the map sending \( i \) to 0, then \( u(i0) \) is the face \( u(0, j) \) of this square. If \((i = j) : i, j \rightarrow j \) 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 preshaf on the slice category \( C^{\text{opp}}/I \). 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 \( Af \) indexed
by \( f : I \to J \) together with a family of restriction maps \( u \mapsto ug, Af \to Afg \) for \( g : J \to K \) such that \( uJ = u \) and \((ug)h = u(gh)\) if \( h : K \to L\). If \( \tau \) is a de Morgan algebra, we can consider \( \tau \) for \( g : J \to K \).

We write \( \tau \) for \( a \) to mean that \( a \) is an element in the set \( A1 \). 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 object 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 \([1]\) is the cubical set defined by \( [1] = dM(J) \). 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 \([0, 1]\) 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 \( i \) 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 \( i \wedge (1 - j) = 0 \) for 0 or 1.)

A face map \( \alpha : I \to I_\alpha \) is a map such that \( \alpha i = i \) and \( \text{dom}(\alpha) = I_{\alpha} = I \). We write \( I_\alpha \) the subset of element \( i \) such that \( \alpha i = i \), and \( \text{dom}(\alpha) = I - I_\alpha \) is the domain of \( \alpha \). If \( \iota_{\alpha} : I_\alpha \to I \) is the inclusion, we have \( \iota_{\alpha} \alpha = 1 \) and hence any face map \( \alpha \) is epi. If \( f : I \to J \) we write \( f \leq \alpha \) to mean that there exists a map \( f' \) (uniquely determined) such that \( f = \alpha f' \). This means that \( i f = \alpha i \) 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 \( fg \leq \alpha \) and \( g \) is strict then \( f \leq \alpha \).

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

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 \( \alpha f = \gamma h = \beta g \). If we write \( \alpha_1 \gamma = \beta \gamma h \) then \( \alpha_1 f = h = \beta_1 g \).


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 that \( f \leq \alpha \) for some \( \alpha \leq L \). We define then \( \beta \leq Lf \) 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 \( L1 = L \) and \((Lf)g = L(fg)\).

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 \( \alpha \alpha_1 = \beta \beta_1 \) then \( a_\alpha \alpha_1 = a_\beta \beta_1 \). We think of such a system as a system of equations \( a\alpha = a_\alpha \) for \( \alpha \) in \( L \). Notice that any element \( v \) in \( A1_I \) defines a compatible system \( a_\alpha = v\alpha \). 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 \( Af \) without ambiguity: if we have both \( f = \alpha g \) and \( f = \beta h \) then \( \alpha \) and \( \beta \) are compatible by Lemma ???. 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 \( Lf \) system \( b_\beta \) by taking \( b_\beta = a_{f\beta} \) and more generally \( b_\beta = a_{fg}. \)

We adopt the following notations for systems. If for instance \( I = i, j \) and \( \vdash I A \) and \( L \) is determined by \((i0) \) and \((j1) \), a L-system \( \bar{d} \) will be determined by an array \((i0) \rightarrow u, (j1) \rightarrow v, \) with \( u \) in \( A(i0) \) and \( v \in A(j1). \) If \( f : I \rightarrow k, l, j \) is defined by \( f(i) = k \land l \) then \( Lf \) is the system \((k0), (l0), (j1) \) and \( \bar{d}f \) is the system \((k0) \rightarrow u, (l0) \rightarrow u, (j1) \rightarrow v. \) If \( g : I \rightarrow j \) is defined by \( g(i) = g(j) = j \) then \( Lg \) is empty and \( \bar{d}g \) is the empty system. Notice that \( \bar{d}(i0)(j1) \) is the system \((i) \rightarrow u(j1) \) which is equal to \((i) \rightarrow v(i0). \)

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 the universe.) The point is to explain how we can function extensionality without using function extensionality at the metalevel.

The syntax for the terms is

\[ t, p, A, E, F ::= x \mid t t \mid \lambda x.t \mid \text{Id} \quad A t t \mid \Pi A F \mid \langle i \rangle t \mid t\vec{p} \mid \text{comp}'(A) \mid t \varphi \]

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 \( \vec{i} \) to represent a system of terms. For instance \( \vec{i} \) may be of the form \((j0) \rightarrow t, (j1) \rightarrow u \) or of the form \((j0) \rightarrow t, (k0) \rightarrow u, (j1)(k1) \rightarrow v. \)

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

\[ (a)(j0) \rightarrow u, \ (j1) \rightarrow v) f = (af)(k0) \rightarrow u, \ (l0) \rightarrow u, \ (k1)(j1) \rightarrow v \]

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

\[ (a)(j0) \rightarrow p, \ (j1) \rightarrow q) (j0) = p \quad (a)(j0) \rightarrow p, \ (j1) \rightarrow q) (j1) = q \]

This operation also satisfies a regularity condition. We have \( a(p, \alpha \rightarrow a\alpha) = a\vec{p}. \) This expresses a strict identity element law for this composition.

We define \( p^* = \langle i \rangle p(1 - i) \) and

\[ \vec{p}a = a\vec{p} \]

The operation \( \text{comp}'(A) \) binds the symbol \( i. \) Its intended type is \( A(i0) \rightarrow A(i1). \) The regularity condition is that \( \text{comp}'(A, a_0) = a_0 \) if \( A \) is independent of \( i. \) We may write \( \text{comp}'(A, a_0) \) instead of \( \text{comp}'(A) a_0. \)
We define \( \text{fill}^i(A, a_0) = \text{comp}^i(A(i \land j), a_0) \) which satisfies \( \Gamma, i : I \vdash \text{fill}^i(A, a_0) : A \) 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 \( \Gamma, i : I \vdash a_0 : A \alpha \) such that \( a_0\alpha = \alpha(i0) \) the element
\[
\text{comp}^i(A, a_0, \bar{a}) = u\bar{a} : A(i1)
\]
where \( u = \text{comp}^i(A, a_0) : A(i1) \) and \( u_\alpha = \langle \bar{i} \rangle \text{comp}^i(A\alpha(i \lor j), a_\alpha) \). This element satisfies
\[
\text{comp}^i(A, a_0, \bar{a})\alpha = a_\alpha(i1)
\]

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

We have the usual \( \beta \)-reduction rule
\[
(\lambda x. t) u = t(x = u)
\]
We write \( (x : A) \rightarrow B \) for \( \Pi A (\lambda x. B) \).

If \( f : I \rightarrow \text{dM}(J) \) we can define the operation \( t \rightarrow tf \) on terms. (Notice that this is a defined operation on terms; we don’t have an explicit term constructor for substitution.) We have
\[
(\langle \bar{i} \rangle t)(\bar{j})f = (\bar{j})tg
\]
where \( g : I, i \rightarrow \text{dM}(J, j) \) extends \( f \) by \( g(i) = j \) not in \( J \). We also have
\[
(\lambda x. t)f = \lambda x. tf \\
xf = x \\
(t u)f = tf uf \\
(t \varphi)f = tf \varphi f \\
(\Pi A F)f = \Pi Af Ff
\]
We can then state the path reduction law
\[
(\langle \bar{i} \rangle t) \varphi = t(i = \varphi)
\]
A canonical object of type \( \text{ld} A a b \) is of the form \( \langle \bar{i} \rangle t \) with \( t(i0) = a \) and \( t(i1) = b \). If \( w \) is of type \( \text{ld} A a b \), then \( w\varphi \) is of type \( A \) and \( w0 = a \) and \( w1 = b \).

If \( L \) is a system for \( I \) we write \( p : \text{ld}^L A u v \) 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 \( a \uparrow \bar{p} \) to be \( \langle \bar{i} \rangle a\bar{q} \) where \( q_\alpha = \langle \bar{j} \rangle p_\alpha(i \land j) \). Using regularity we have that \( a \uparrow \bar{p} \) defines a line \( l : a \sim (\bar{q}\bar{p}) \) such that \( la = p_\alpha \). We also define \( \bar{p} \uparrow a = a \uparrow \bar{p}^\prime \).

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
\[
(w\bar{p}) a = (w a)\bar{q}
\]
where \( q_\alpha = \langle \bar{i} \rangle p_\alpha(i, a\alpha) \).

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 [?, ?].

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

For identity types, we have
\[(p \bar{q}) \varphi = p \varphi^r\]
where \(r_\alpha = (i)q_\alpha(i, \varphi).

We also have
\[\text{comp}^i(\text{Id} A a b, p_0) = (j)\text{comp}^j(A, \bar{a}, p_0 j)\]
\(\bar{a}\) 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 \(\text{comp}^i(B(p i), b_0)\) 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
\[q = (i)(p(i), (j)p(i \land j))\]
is a path such that \(q(0) = (a, (j)a)\) and \(q(1) = (x, (j)p(j)) = (x, p)\). If \(x = a\) and \(p = (i)a\) (which interprets reflexivity) we get \(q = (i)(a, (j)a)\) 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 \(\Gamma \vdash, \Gamma \vdash A\) and \(\Gamma \vdash t : A\) with
\[
\Gamma, \Delta ::= () \mid \Gamma, x : A \mid \Gamma, i : \mathbb{I}
\]
The rules are the usual rules of type theory. We can consider the set of symbols in a context \(\Gamma\) and define \(\Gamma \alpha\) if \(\alpha\) is a face of \(I\). The rule
\[
\frac{\Gamma \vdash a : A}{\Gamma \alpha \vdash a\alpha : A\alpha}
\]
is admissible.

The new rules are then the following.
\[
\frac{\Gamma \vdash a : A}{\Gamma \vdash a\alpha : A\alpha}
\]
with a computation rule \((a\bar{p})\alpha = u_\alpha\). We also have
\[
\frac{\Gamma \vdash A}{\Gamma \vdash \text{Id} A a_0 a_1}
\frac{\Gamma \vdash A}{\Gamma \vdash i : \mathbb{I} \vdash t : A}
\frac{\Gamma \vdash (i)t : \text{Id} A I(0) I(1)}
\]
In particular, we get the reflexivity proof of \(a : A\) as the constant path \((i)a\)
\[
\frac{\Gamma \vdash t : \text{Id} A a_0 a_1}{\Gamma \vdash t \varphi : \mathbb{I}}
\frac{\Gamma \vdash t : \text{Id} A a_0 a_1}{\Gamma \vdash t 0 = a_0 : A}
\frac{\Gamma \vdash t : \text{Id} A a_0 a_1}{\Gamma \vdash t 1 = a_1 : A}
\]
We also have
\[
\frac{\Gamma \vdash i : I \vdash A}{\Gamma \vdash \text{comp}^i(A) : A(I(0)) \rightarrow A(I(1))}
\]
We can justify function extensionality by the defined constant
\[
\frac{\Gamma \vdash t : (x : A) \rightarrow B}{\Gamma \vdash u : (x : A) \rightarrow B}
\frac{\Gamma \vdash p : (x : A) \rightarrow \text{Id} B (t x) (u x)}{\Gamma \vdash \text{ext} t u p : \text{Id} ((x : A) \rightarrow B) t u}
\]
where \(\text{ext} t u p = (i)\lambda x.p x i\).
6 Denotational semantics

A context is interpreted by a cubical set. A judgement \( \Gamma \vdash A \) is interpreted by giving a set \( A^\rho \) for each \( \rho \) in \( \Gamma(I) \) with restriction maps \( u \mapsto uf \). \( A^\rho \rightarrow A^\rho f \) for \( f : I \rightarrow J \) satisfying \( uf = u \) and \( uf g = u(fg) \). Furthermore this should have composition and transport operations.

For composition, we should have an operation \( u|_i \overline{u} \) in \( A^\rho \) for \( u \) in \( A^\rho \) and \( u_\alpha \) in \( A^\rho \alpha_i \) is a compatible family such that \( u|_i u_\alpha = u_\alpha(i) \). This operation should be regular and uniform. The regularity is that \( u|_i \overline{u} = u|_i \overline{v} \) if \( \overline{v} \) is obtained from \( \overline{u} \) by taking away some \( u_\alpha \) independent of \( i \). The uniformity is that \( (u|_i \overline{u})f = uf|_i \overline{u}(f, i = j) \) if \( f : I \rightarrow J \) and \( j \) not in \( J \).

For transport, we should have an operation \( \text{comp}^j(u) \) in \( A^\rho(j1) \) if \( j \) in \( J \) and \( u \) in \( A^\rho(j0) \). This operation should be regular: if \( \rho \) is independent of \( j \), i.e. \( \rho = \rho(j0)v_j \), then \( \text{comp}^j(u) = u \) and uniform: \( \text{comp}^j(A^\rho, u)f = \text{comp}^j(A^\rho, uf, \alpha) \) if \( f : I \rightarrow J \) and \( k \) is not in \( J \).

7 Equivalence

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

7.1 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 correspondind functions.

7.2 Graduate Lemma

Lemma 7.1 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 7.2 If \( E : T \rightarrow I \) then \( \text{comp}^i(E) \) is an equivalence.

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

We consider only presheaves \( A \) on \( \mathcal{C}\text{opp} \) 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 \( \mathcal{C} \) then \( \mathcal{G}^F(I) \) is a set of sequences \( \lambda_f \) in \( F(J) \rightarrow G(J) \) for \( f : I \rightarrow J \), satisfying the condition \( (\lambda_f u)g = \lambda_f ug \), and hence \( \lambda \) can be written as a tuple \( (\lambda_\alpha) \) with \( \lambda_\alpha = (\lambda_\alpha \beta_1) \) indexed by \( \text{strict} \) map \( g : I_\alpha \rightarrow J \). 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 \( \vdash_{J_0} \sigma_\alpha : T_\alpha \to A_\alpha \) we define a new type \( \vdash_I B = \bar{\sigma}|A \). For \( f : I \to J \), an element of \( Bf \) is a \( J \)-element \((\bar{t}, u)\) where \( u \) is in \( Af \) and we have \( \sigma_{\beta\bar{t}} = u\beta \) 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 \textit{glueing} (which transforms equivalence to equality).

If we have \( B = \bar{\sigma}|A \) then there is a canonical map \( \delta : B \to A \). For instance, if \( I = i \) and we have \( \sigma_{i0} : T_{i0} \to A_{i0} \) then \( B \) is the set of sequences \((t_{i0}, a_{i1}, a_1)\) such that \( a_1 \) is in \( A_1(\sigma_{i0} t_{i0}, a_{i1}) \) and we define \( \delta(t_{i0}, a_{i1}, a_1) = (\sigma_{i0} t_{i0}, a_{i1}, a_1) \).

## 9 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 9.1** If we have \( \vdash_I \sigma : T \to A \) then we have a path \( \sigma(i1) (\text{comp}^i(T, t_0)) \to \text{comp}^i(A, \sigma(i0) t_0) \).

*Proof.* We define a system \( \bar{w} \) as \((j0) \mapsto \sigma(\text{fill}^i(T, t_0)), (j1) \mapsto \text{fill}^i(A, \sigma(i0) t_0) \) and \( p = \langle j \rangle \text{comp}^i(A, \bar{w}, a_0) \) is a path \( u(i1) \to v(i1) \).

**Lemma 9.2** If we have \( \vdash_I \sigma : T \to A \) then for any system of paths \( \bar{p} \) compatible with \( t : T \) we have \( \sigma(tp) \sim_L \sigma t \bar{q} \) where \( q_0 = \langle i \rangle \sigma \alpha p_\alpha(i) \).

Given a \( L \)-system \( \bar{T} \) 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

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

As a cubical set \( B \) is \( \bar{\sigma}A \) where \( \sigma_\alpha = u_\alpha \). An element of this type is of the form \((\bar{t}, a)\) with \( t_\alpha : T_\alpha \) and \( a : A \) such that \( a\alpha = \sigma_\alpha t_\alpha \). If \( L = () \), we only have one equivalence \( \sigma(.) : T_1 \to A \) and \( B = T_1 \). If \( L \) is empty we have \( B = A \). If \( f : I \to J \) we define

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

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 \( \delta(\bar{t}, a) = a \) and \( \delta at = \sigma_\alpha t \) for \( \alpha \) in \( L \).

**Proposition 9.3** The type \( B = A \bar{\bar{u}} \) has composition operations.

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

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

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

We should have, for each \( \alpha \in L \), \( t_\alpha = v_1 \alpha = v_0 \alpha q_\alpha : T_\alpha \). On the other hand, we have \( a_1 \alpha = \alpha a_0 \) in \( A_\alpha \).

Using Lemma ??, we have that \( a_1 \alpha \sim_J (A_\alpha) \sigma_\alpha t_\alpha \) and so, we have a line \( r_\alpha : a_1 \alpha \to \sigma_\alpha t_\alpha \) constant on \( J_\alpha \). We define \( v_1 = (\bar{t}, a_1 \bar{r}) \).

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

**Proposition 9.4** The type \( B = A \bar{\bar{u}} \) has transport functions.
Proof. We have to define \( \text{comp}'(B, v_0) : B(i) \) for \( v_0 : B(i0) \). We define \( a_0 = \delta(i0) \) \( v_0 \) and \( a_1 = \text{comp}'(A, a_0) \).

Let \( L' \) be the subset of \( \gamma \) in \( L \) not mentioning \( i \) and \( L'' \) subset of \( \gamma \) such that \( \gamma(i1) \) is 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 \( L(i) \) is a union of \( L'' \) and of \( L_1 \) 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 \( u_1 \gamma = \sigma_\gamma(i1) t_\gamma \) where \( t_\gamma = \text{comp}'(\gamma, v_\gamma, v_\gamma) \) is of type \( T_\gamma(i1) \).

For \( \gamma \) in \( L'' \), then \( u_1 \gamma \) should be of the form \( \sigma_\gamma(i1) \rho_\gamma \) for some \( \rho_\gamma \) in \( T_\gamma(i1) \).

We first deal with the constraints for \( \gamma \) in \( L' \). We have

\[
\begin{align*}
a_0 \gamma &= \sigma_\gamma(i0) v_0 \\
a_1 \gamma &= \text{comp}'(\gamma, a_0 \gamma) \\
t_\gamma &= \text{comp}'(\gamma, v_0 \gamma)
\end{align*}
\]

We can build a line \( w_\gamma : \sigma_\gamma(i1) t_\gamma \to a_1 \gamma \) using Lemma ??.

(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 \( a_i' = u_0 a_i \). We have \( a_i' \gamma = \sigma_\gamma(i1) t_i \) 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 \( L' \gamma \) we have \( \gamma \beta \leq L' \) and we also have \( a_i' \gamma \beta = \sigma_\beta a_i \beta_{\gamma} \) 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 \( s_\gamma : \sigma_\gamma t_\gamma \to a_i' \gamma \) which is a \( L' \gamma \)-path. We change then \( a_i' \) to \( u_1 = s_\alpha a_i' \).

By regularity, we have \( u_1 \delta = a_i' \delta \) for \( \delta \) in \( L' \), so we did not modify the \( L' \) faces of \( a_i' \).

The element \( u_1 = s_\alpha a_i' \) satisfies \( u_1 \gamma = \sigma_\gamma t_\gamma \) for \( \gamma \) in \( L' \) and \( u_1 \gamma = \sigma_\gamma(t_\gamma) \gamma \) for \( \gamma \) in \( L'' \). Hence, we can define a corresponding element \( v_1 = (\vec{r}, u_1) \) 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 \( L'' \).

The corresponding typing rules are

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

\[
\frac{\Gamma \vdash a : A \quad \Gamma \vdash u_\alpha : \text{iso}(A \alpha, T_\alpha) \quad \Gamma \vdash u_\alpha \gamma \alpha = a \alpha : A \alpha}{\Gamma \vdash (t, a) : A \vec{u}}
\]

We also have

\[
\frac{\Gamma \vdash b : A \vec{u}}{\Gamma \vdash (\vec{v}, b) : A}
\]

where \( v_\alpha = u_\alpha \) \( \text{box} \).

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} \( \phi \) where \( \phi \) 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 \( S^1(I, \alpha, i) \), and such that \( u_\alpha(i0) = u_\alpha \alpha \).

Thus the element of \( S^1(I) \) are defined by these generators together with the relation that we have \( \text{comp}'(\vec{a}, u_0) = u_0 \) 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 \( \vdash \text{loop} : S^1 \). 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 \( \vdash l : F \langle \text{loop} i \rangle \). Furthermore, we have \( \vdash s \text{ base} = b : F \text{ base} \) and \( \vdash l : F \langle \text{loop} i \rangle = l : F \langle \text{loop} i \rangle \).
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 \( \Gamma A \). Concretely \( A \) is a family of small sets \( Af \) with restriction maps \( Af \to Af g, u \mapsto uf \) satisfying \( u1 = u \) and \( u(f)g = u(fg) \). 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 \( Af \)
2. or \( \text{squash} \varphi \ u0 \ u1 \) with \( \varphi \) in \( \text{dM}(J) \) and \( u0, u1 \) in \( \text{inh}(A)f \) and \( \varphi \neq 0, 1 \)
3. or of the form \( u0 \bar{p} \) where \( u0 \) is in \( \text{inh}(A)f \) and \( p_{\alpha} \) is in \( \text{Path}(\text{inh}(A))f_{\alpha} \)

One needs also to take care of the regularity condition, so that \( u0 \bar{p} = u0 \bar{q} \) if \( \bar{q} \) is obtained from \( \bar{p} \) by removing some element that are constant. 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 [?].)

We define \( \text{comp}^i((v_0) \in \text{inh}(A)f(j1)) \) if \( v_0 \) is in \( \text{inh}(A)f(j0) \). Let \( tr \) be the transport function \( v \mapsto \text{comp}^i(v_0) \). The definition of \( tr(v_0) \) is done by induction on \( v_0 \):

1. in the case where \( v_0 \) is in the form \( \text{inc}(u_0) \) we only need that \( A \) has transport
2. in the case where \( v_0 \) is of the form \( \text{squash} \varphi \ u0 \ u1 \) then \( tr(v_0) \) is \( \text{squash} \varphi \ tr(\alpha) \ tr(\beta) \)
3. in the case of \( v_0 \) is of the form \( \text{comp}^i(\alpha, u_0) \) where \( j \) not in \( J \) then \( tr(v_0) \) is \( \text{comp}^i(tr(\alpha), tr(\beta)) \)

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 \( A \) 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 \( u0 \bar{p} \) where \( u0 \) is in \( Af \) 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: 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 \( \mathbb{I}(J) = \text{dM}(J) \) 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 \( \mathbb{I}([i]) \to N \).

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 \( f : A \rightarrow B \) if we have \( a : A \) and \( b : B \) and a path \( f \ a \rightarrow b \) then we can find \( g : A \rightarrow B \) such that \( g \ a = b \) with a path \( f \rightarrow g \).

Indeed we can consider the trivial fibration \((y : B, \text{id} B (f \ x) y)\), \( x : A \) and the monomorphism \( 1 \rightarrow A \) defined by \( a : A \). We have a map \((b, q) : 1 \rightarrow (y : B, \text{id} B (f \ a) y)\). If we had the right lifting property we could find a lifting map \((x : A) \rightarrow (y : B, \text{id} B (f \ x) y), x \mapsto (g \ x, p \ x)\) 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 \( u \rightarrow b \) and \( a' \) at time 0 and only one \( u' = u \rightarrow b \) at time 1. We then have a map \( f : A \rightarrow B \) taking \( f \ a = u \) and \( f \ a' = u' \) and we have a path \( f \ a \rightarrow b \), but there is no map \( g : A \rightarrow B \) such that \( g \ a = b \) with a path \( f \rightarrow g \). (Notice that this counter-example holds already in the strict groupoid model: already for this model, we cannot have a constructive model structure.)

Appendix 4: New Typing Rules

We collect all new rules of our system.

\[
\Gamma \vdash a : A \quad \Gamma \vdash p_\alpha : \text{id} A \alpha a \alpha u_\alpha \\
\frac{}{\Gamma \vdash a \alpha \overline{p} : A} \\
\frac{\Gamma \vdash A \quad \Gamma \vdash a_0 : A \quad \Gamma \vdash a_1 : A \quad \Gamma \vdash \text{id} A \ a_0 \ a_1}{\Gamma \vdash t : \text{id} A \ a_0 \ a_1} \\
\frac{\Gamma \vdash \varphi : \mathbb{I}}{\Gamma \vdash t \varphi : A} \\
\frac{\Gamma \vdash i \ : \mathbb{I} \vdash A}{\Gamma \vdash \text{comp}^i(A) : A(i0) \rightarrow A(i1)} \\
\frac{\Gamma \vdash A \quad \Gamma \vdash u_\alpha : \text{iso}(A\alpha, T_\alpha)}{\Gamma \vdash A\overline{\alpha} u} \\
\frac{\Gamma \vdash a : A \quad \Gamma \vdash u_\alpha : \text{iso}(A\alpha, T_\alpha) \quad \Gamma \vdash u_\alpha t_\alpha = a \alpha : A\alpha}{\Gamma \vdash (t, a) : A\overline{\alpha}} \\
\frac{\Gamma \vdash b : A\overline{\alpha}}{\Gamma \vdash (\overline{\alpha}, b) : A} \quad \text{where} \ v_\alpha = u_\alpha b\alpha \\
\frac{\Gamma \vdash A \quad \Gamma \vdash E_\alpha : \text{id} U A\alpha T_\alpha}{\Gamma \vdash A\overline{E}} \\
\frac{\Gamma \vdash a : A \quad \Gamma \vdash E_\alpha : \text{id} U A\alpha T_\alpha \quad \Gamma \vdash \text{comp}^i(E_\alpha^\ast, t_\alpha) = a \alpha : A\alpha}{\Gamma \vdash (t, a) : A\overline{E}}
\]

References


