Cubical Type Theory

May 4, 2015

Interval

$$\varphi, \psi \ ::= \ 0 \mid 1 \mid i \mid 1-i \mid \varphi \wedge \psi \mid \varphi \vee \psi$$

The equality is the equality in the free distributive lattice on generators $i, 1-i$. We don't get a Boolean algebra since we don't require neither $i \wedge (1-i) = 0$ nor $i \vee (1-i) = 1$.

Context

$$\Delta, \Gamma \ ::= \ () \mid \Gamma, x : A \mid \Gamma, i : \mathbb{I}$$

Substitutions

$$\frac{}{() : \Delta \to ()} \qquad \frac{\sigma : \Delta \to \Gamma \qquad \Delta \vdash u : A\sigma}{(\sigma, x = u) : \Delta \to \Gamma, x : A} \qquad \frac{\sigma : \Delta \to \Gamma \qquad \Delta \vdash \varphi : \mathbb{I}}{(\sigma, i = \varphi) : \Delta \to \Gamma, i : \mathbb{I}}$$

$$\frac{\sigma : \Delta \to \Gamma \quad \Gamma \vdash A}{\Delta \vdash A\sigma} \qquad \frac{\sigma : \Delta \to \Gamma \quad \Gamma \vdash t : A}{\Delta \vdash t\sigma : A\sigma}$$

We can define $1_\Gamma : \Gamma \to \Gamma$ by induction on Γ and then if $\Gamma \vdash u : A$ we write $(x = u) : \Gamma \to \Gamma, x : A$ for $1_\Gamma, x = u$. If we have further $\Gamma, x : A \vdash t : B$ we may write $t(u)$ and $B(u)$ respectively instead of $t(x = u)$ and $B(x = u)$.

Similarly if $\Gamma \vdash \varphi : \mathbb{I}$ we write $(i = \varphi) : \Gamma \to \Gamma, i : \mathbb{I}$ for $1_\Gamma, i = \varphi$. We may write $t(\varphi)$ and $B(\varphi)$ for $t(i = \varphi)$ and $B(i = \varphi)$ respectively if $\Gamma, i : \mathbb{I} \vdash t : B$.

Face operations and Notation for systems

Among these substitutions, there are the ones corresponding to face operations e.g.

$$(x = x, i = 0, y = y) : (x : A, y : B(i = 0)) \to (x : A, i : \mathbb{I}, y : B)$$

If Γ is $(x : A, i : \mathbb{I}, y : B)$ we write $\Gamma(i0) = (x : A, y : B(i0))$ and we write simply $(i0) : \Gamma(i0) \to \Gamma$ instead of $(x = x, i = 0, y = y)$. In general, we write $\alpha : \Gamma\alpha \to \Gamma$ the face operations.

A substitution $\sigma : \Delta \to \Gamma$ is *strict* if it never takes the value $0, 1$ on symbols. A fundamental fact is that any substitution σ is decomposed in a unique way in the form $\sigma = \alpha\sigma_1$ where σ_1 is strict.

A *system* for a type $\Gamma \vdash A$ is given by a set of compatible objects $\Gamma\alpha \vdash u_\alpha : A\alpha$.

Proposition 0.1 *If we have $\sigma : \Delta \to \Gamma\alpha$ and $\delta : \Delta \to \Gamma\beta$ such that $\alpha\sigma = \beta\delta : \Delta \to \Gamma$ then $\Delta \vdash u_\alpha\sigma = u_\beta\delta : A\alpha\sigma$.*

Proof. α and β are compatible and we can find β_1, α_1 such that $\alpha\beta_1 = \beta\alpha_1 = \gamma$ and $\sigma = \beta_1\sigma_1$, $\delta = \alpha_1\sigma_1$ and then $u_\alpha\sigma = u_\alpha\beta_1\sigma_1 = u_\beta\alpha_1\sigma_1$. □

In order to write the equation for transport and composition, it is appropriate to use the following notation $[\alpha \mapsto a_\alpha]$ for a system $\vec{a}$ in A. Let L be the family of faces over which the system $\vec{a}$ is defined. If $\sigma : \Delta \to \Gamma$, we write $\sigma \leqslant L$ if, and only if, we can write $\sigma = \alpha\sigma_1$ for some α in L. The previous result shows that in this case $\vec{a}\sigma = a_\alpha\sigma_1$ is defined without ambiguity.

Given L a set of face maps $\Gamma\alpha \to \Gamma$ The set of all maps $\sigma : \Delta \to \Gamma$ such that $\sigma \leqslant L$ is a *sieve* on Γ: if $\sigma \leqslant L$, then $\sigma\delta \leqslant L$.

Lemma 0.2 *If δ is strict and $\sigma\delta \leqslant L$ then $\sigma \leqslant L$*

Proof. We have $\sigma\delta = \alpha\theta$ for some α in L. Hence we have $i\sigma\delta = i\alpha$ for all i symbol declared in Γ. Since δ is strict this implies $i\sigma = i\alpha$. □

A sieve is actually determined by the face maps it contains. This follows directly from the previous Lemma and the fact that any map σ can be written $\alpha\sigma_1$ with σ_1 strict.

Given L a downward closed set of face maps on Γ and $\sigma : \Delta \to \Gamma$ we define $L\sigma$ to be the downward closed set of face maps β on Δ such that $\sigma\beta \leqslant L$.

Corollary 0.3 *We have $\delta \leqslant L\sigma$ if, and only if, $\sigma\delta \leqslant L$*

Proof. We write $\delta = \beta\delta_1$ where δ_1 is strict and we have $\sigma\delta = \sigma\beta\delta_1 \leqslant L$ if, and only if, $\sigma\beta \leqslant L$ by the Lemma. □

Corollary 0.4 *We have $L1 = L$ and $(L\sigma)\delta = L(\sigma\delta)$*

If now $\sigma : \Delta \to \Gamma$ is arbitrary, we can define $\vec{a}\sigma$ as the system $[\beta \mapsto \vec{a}\sigma\beta]$ for β such that $\sigma\beta \leqslant L$. This defines a system for $\Delta \vdash A\sigma$. It follows from this Corollary and from Proposition 0.1 that we have $(\vec{u}\sigma)\delta = \vec{u}(\sigma\delta)$.

If $f : A \to B$ and $\vec{a}$ is a system for A we define $f\ \vec{a} = [\alpha \mapsto f\alpha\ a_\alpha]$ which is a system for B.

Basic typing rules

$$\frac{\Gamma \vdash A}{\Gamma, x : A \vdash} \qquad \frac{\Gamma \vdash}{\Gamma, i : \mathbb{I} \vdash}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash x : A}\ (x : A \ in\ \Gamma) \qquad \frac{\Gamma \vdash}{\Gamma \vdash i : \mathbb{I}}\ (i : \mathbb{I}\ in\ \Gamma)$$

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

Sigma types

$$\frac{\Gamma, x : A \vdash B}{\Gamma \vdash (x : A, B)} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B(a)}{\Gamma \vdash (a, b) : (x : A, B)} \qquad \frac{\Gamma \vdash z : (x : A, B)}{\Gamma \vdash z.1 : A} \qquad \frac{\Gamma \vdash z : (x : A, B)}{\Gamma \vdash z.2 : B(z.1)}$$

Identity types

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash \mathsf{ID}\ A\ B} \qquad \frac{\Gamma, i : \mathbb{I} \vdash A}{\Gamma \vdash \langle i \rangle A : \mathsf{ID}\ A(i = 0)\ A(i = 1)}$$

$$\frac{\Gamma \vdash P : \mathsf{ID}\ A_0\ A_1 \qquad \Gamma \vdash \varphi : \mathbb{I}}{\Gamma \vdash P\ \varphi} \qquad \frac{\Gamma \vdash P : \mathsf{ID}\ A_0\ A_1}{\Gamma \vdash P\ 0 = A_0} \qquad \frac{\Gamma \vdash P : \mathsf{ID}\ A_0\ A_1}{\Gamma \vdash P\ 1 = A_1}$$

$$\frac{\Gamma \vdash P : \mathsf{ID}\ A_0\ A_1 \qquad \Gamma \vdash a_0 : A_0 \qquad \Gamma \vdash a_1 : A_1}{\Gamma \vdash \mathsf{IdP}\ P\ a_0\ a_1} \qquad \frac{\Gamma, i : \mathbb{I} \vdash t : A}{\Gamma \vdash \langle i \rangle t : \mathsf{IdP}\ (\langle i \rangle A)\ t(i0)\ t(i1)}$$

$$\frac{\Gamma \vdash t : \mathsf{IdP}\ P\ a_0\ a_1 \quad \Gamma \vdash \varphi : \mathbb{I}}{\Gamma \vdash t\ \varphi : P\ \varphi} \qquad \frac{\Gamma \vdash t : \mathsf{IdP}\ P\ a_0\ a_1}{\Gamma \vdash t\ 0 = a_0 : P\ 0} \qquad \frac{\Gamma \vdash t : \mathsf{IdP}\ P\ a_0\ a_1}{\Gamma \vdash t\ 1 = a_1 : P\ 1}$$

We define $\mathsf{Id}\ A\ a_0\ a_1 = \mathsf{IdP}\ (\langle i\rangle A)\ a_0\ a_1$ if $a_0 : A$ and $a_1 : A$. We can define $1_a : \mathsf{Id}\ A\ a\ a$ as $1_a = \langle i\rangle a$.

We define $p^* = \langle i\rangle p\ (1-i)$ so that

$$\frac{\Gamma \vdash p : \mathsf{Id}\ A\ a\ b}{\Gamma \vdash p^* : \mathsf{Id}\ A\ b\ a}$$

With these rules we also can justify function extensionality

$$\frac{\Gamma \vdash t : (x:A) \to B \qquad \Gamma \vdash u : (x:A) \to B \qquad \Gamma \vdash p : (x:A) \to \mathsf{Id}\ B\ (t\ x)\ (u\ x)}{\Gamma \vdash \langle i\rangle \lambda x : A.\ p\ x\ i : \mathsf{Id}\ ((x:A) \to B)\ t\ u}$$

We also can justify the fact that any element in $(x : A, \mathsf{Id}\ A\ a\ x)$ is equal to $(a, 1_a)$

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

For justifying the transitivity of equality, we need A to have *composition operations*.

Composition operations

We have

$$\frac{\Gamma \vdash a : A \qquad \Gamma\alpha \vdash p_\alpha : \mathsf{Id}\ A\alpha\ a\alpha\ u_\alpha}{\Gamma \vdash \mathsf{comp}\ A\ a\ \vec{p} : A}$$

with the uniformity condition, for $\sigma : \Delta \to \Gamma$

$$\Delta \vdash (\mathsf{comp}\ A\ a\ \vec{p})\sigma = \mathsf{comp}\ A\sigma\ a\sigma\ \vec{p}\sigma : A\sigma$$

and the regularity condition

$$\Gamma \vdash \mathsf{comp}\ A\ a\ (\vec{p}, \alpha \mapsto \langle i\rangle a\alpha) = \mathsf{comp}\ A\ a\ \vec{p} : A$$

We may write simply $a\vec{p}$ instead of $\mathsf{comp}\ A\ a\ \vec{p}$.

We can then justify

$$\frac{\Gamma \vdash p : \mathsf{Id}\ A\ a\ b \qquad \Gamma \vdash q : \mathsf{Id}\ A\ b\ c}{\Gamma \vdash \langle i\rangle (p\ i)[(i=1) \mapsto q] : \mathsf{Id}\ A\ a\ c}$$

With such a composition operation, each type has the structure of a weak ∞-groupoid.

If we define

$$\mathsf{prop}\ A = (x\ y : A) \to \mathsf{Id}\ A\ x\ y \qquad \mathsf{set}\ A = (x\ y : A) \to \mathsf{prop}\ (\mathsf{Id}\ A\ x\ y)$$

it is possible to show that any proposition is a set as follows

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

Transport operation

$$\frac{\Gamma, i : \mathbb{I} \vdash A}{\Gamma \vdash \mathsf{transp}^i(A) : A(i0) \to A(i1)}$$

together with the regularity condition that $\mathsf{transp}^i(A)\ a_0 = a_0$ whenever A is independent of i.

We can then justify the substitution rule

$$\frac{\Gamma, x : A \vdash B \qquad \Gamma \vdash p : \mathsf{Id}\ A\ a\ b}{\Gamma \vdash \mathsf{transp}^i(B(p\ i)) : B(a) \to B(b)}$$

which, together with the fact that any type $(x : A, \mathsf{Id}\ A\ a\ x)$ is contractible, implies the usual dependent elimination rule for the identity type.

If $E : \mathsf{ID}\ A\ B$ we write $E^+ = \mathsf{transp}^i(Ei) : A \to B$ and $E^- = \mathsf{transp}^i(E(1-i)) : B \to A$.

Kan filling operation

It is convenient for the definition of composition to introduce the operation $\mathsf{comp}^i\ A\ a\ \vec{a} : A$ with $i : \mathbb{I} \vdash a_\alpha : A\alpha$ compatible system such that $a_\alpha(i0) = a\alpha$. We have $(\mathsf{comp}^i\ A\ a\ \vec{a})\alpha = a_\alpha(i1)$. This operation binds the symbol i.

The composition operation can then be defined as $\mathsf{comp}\ A\ a\ \vec{p} = \mathsf{comp}^i\ A\ a\ [\alpha \mapsto p_\alpha\ i]$

In general a map $u : T \to A$ does not need to preserve composition for judgemental equality. However, if we have a map $u : T \to A$, for any $t : T$ and system $i : \mathbb{I} \vdash t_\alpha : T\alpha$ we can consider the composition of the images $v_0 = \mathsf{comp}^i\ A\ (u\ t)\ (u\ \vec{t})$ and the image of the composition $v_1 = u\ (\mathsf{comp}^i\ T\ t\ \vec{t})$ and we have an equality

$$\vdash \mathsf{pres}\ u\ \vec{t} : \mathsf{Id}\ A\ v_0\ v_1$$

which satisfies $(\mathsf{pres}\ u\ \vec{t})\alpha = \langle i\rangle(u\alpha\ t_\alpha(i1))$ for $\alpha \leqslant L$.

This is defined as follow. First we consider $w_0 = \mathsf{fill}^i\ A\ (u\ t)\ (u\ \vec{t})$ and $w_1 = u\ (\mathsf{fill}^i\ T\ t\ \vec{t})$. We have $w_0(i0) = u\ t$ and $w_0(i1) = v_0$, $w_0\alpha = u\ t_\alpha$ while $w_1(i0) = u\ t$ and $w_1(i1) = v_1$, $w_1\alpha = w_0\alpha = u\ t_\alpha$. We then take

$$\mathsf{pres}\ u\ \vec{t} = \langle j\rangle(\mathsf{comp}^i\ A\ (u\ t)\ [\alpha \mapsto u\alpha\ t_\alpha, (j=0) \mapsto w_0, (j=1) \mapsto w_1])$$

This operation satisfies $(\mathsf{pres}\ u\ \vec{t})\sigma = \mathsf{pres}\ u\sigma\ \vec{t}\sigma$.

We recover Kan filling operation

$$i : \mathbb{I} \vdash \mathsf{fill}^i\ A\ a\ \vec{a} = \mathsf{comp}^j\ A\ a\ [\alpha \mapsto a_\alpha(i \wedge j)] : A$$

The element $i : \mathbb{I} \vdash u = \mathsf{fill}^i\ A\ a\ \vec{a} : A$ satisfies $u(i0) = a : A$ and $u(i1) = \mathsf{comp}^i\ A\ a\ \vec{a} : A$.

Recursive definition of composition

The operation $\mathsf{comp}^i\ A\ a\ \vec{a}$ is defined by induction on A.

Product type

In the case of a product type $\vdash (x : A) \to B = C$, we have a system $i : \mathbb{I} \vdash \mu_\alpha : C\alpha$ with $\mu_\alpha(i0) = f\alpha$ and we define

$$\mathsf{comp}^i\ C\ f\ \vec{\mu} = \lambda x : A.\ \mathsf{comp}^i\ B\ (f\ x)\ [\alpha \mapsto \mu_\alpha\ x] : C$$

Identity type

In the case of identity type $\vdash \mathsf{Id}\ A\ u\ v = C$ if we have a system $i : \mathbb{I} \vdash \mu_\alpha : C\alpha$ with $\mu_\alpha(i0) = p\alpha$ for $p : C$. We define

$$\mathsf{comp}^i\ C\ p\ \vec{\mu} = \langle j\rangle \mathsf{comp}^i\ A\ (p\ j)\ [\alpha \mapsto \mu_\alpha\ j] : C$$

Sum type

In the case of a sigma type $\vdash (x : A, B) = C$ we first need to generalize the composition operation $\mathsf{Comp}^i\ A\ a\ \vec{a} : A(i1)$ where A may now depend on $i : \mathbb{I}$ and $\vdash a : A(i0)$. This is defined in term of composition *and* transport operations.

$$\mathsf{Comp}^i\ A\ a\ \vec{a} = \mathsf{comp}^i\ A(i1)\ (\mathsf{transp}^i\ A\ a)\ [\alpha \mapsto \mathsf{transp}^j\ A\alpha(i \vee j)\ a_\alpha] : A(i1)$$

Given a system $\vec{c} = [\alpha \mapsto (a_\alpha, b_\alpha)]$ for C, we define

$$\mathsf{comp}^i\ C\ (a,b)\ \vec{c} = (\mathsf{comp}^i\ A\ a\ [\alpha \mapsto a_\alpha], \mathsf{Comp}^i\ B(u)\ b\ [\alpha \mapsto b_\alpha])$$

where $u = \mathsf{comp}^j\ A\ a\ [\alpha \mapsto a_\alpha(i \wedge j)]$.

Recursive definition of transport

The operation $\mathsf{transp}^i\ A\ a$ is defined by induction on A.

Product type

In the case of a product type $\vdash (x : A) \to B = C$, we define

$$\mathsf{transp}^i\ C\ f = \lambda x : A(i1).\ \mathsf{transp}^i\ B(u)\ (f\ (\mathsf{transp}^i\ A(1-i)\ x))\ \ : C(i1)$$

where $x : A(i1), i : \mathbb{I} \vdash u = \mathsf{transp}^j\ A(i \vee 1-j)\ x : A$.

Identity type

In the case of identity type $\vdash \mathsf{Id}\ A\ u\ v = C$ we define

$$\mathsf{transp}^i\, C\, p = \langle j\rangle \mathsf{comp}^i\ A(i1)\ (\mathsf{transp}^i\ A\ (p\ j))\ [(j=0) \mapsto \mathsf{transp}^k\ A\ u(i{\vee}k), (j=1) \mapsto \mathsf{transp}^k\ A\ v(i{\vee}k)] : C(i1)$$

Sum type

In the case of a sigma type $\vdash (x : A, B) = C$, we define

$$\mathsf{transp}^i\ C\ (a, b) = (\mathsf{transp}^i\ A\ a, \mathsf{transp}^i\ B(u)\ b)$$

where $i : \mathbb{I} \vdash u = \mathsf{transp}^j\ A(i \wedge j)\ a$.

Isomorphisms

We define the type of isomorphisms

$$\frac{\Gamma \vdash f : A \to B \quad \Gamma \vdash g : B \to A \quad \Gamma \vdash s : (y : B) \to \mathsf{Id}\ B\ (f\ (g\ y))\ y \quad \Gamma \vdash t : (x : A) \to \mathsf{Id}\ A\ (g\ (f\ x))\ x}{\Gamma \vdash (f, g, s, t) : \mathsf{Iso}(A, B)}$$

We write $(f, g, s, t)^+ = f : A \to B$ and $(f, g, s, t)^- = g : B \to A$.

Glueing

$$\frac{\Gamma \vdash A \quad \Gamma\alpha \vdash u_\alpha : \mathsf{Iso}(A\alpha, T_\alpha)}{\Gamma \vdash A\vec{u}}$$

$$\frac{\Gamma \vdash a : A \quad \Gamma\alpha \vdash u_\alpha : \mathsf{Iso}(A\alpha, T_\alpha) \quad \Gamma\alpha \vdash u_\alpha^- t_\alpha = a\alpha : A\alpha}{\Gamma \vdash (\vec{t}, a) : A\vec{u}}$$

$$\frac{\Gamma \vdash A \quad \Gamma\alpha \vdash u_\alpha : \mathsf{Iso}(A\alpha, T_\alpha)}{\Gamma \vdash \mathsf{elim}\ A\ \vec{u} : A\vec{u} \to A}$$

We write $B = A\vec{u}$. We have $B\alpha = T_\alpha$ for $\alpha \leqslant L$. We have a map $\vdash g = \mathsf{elim}\ A\ \vec{u} : B \to A$ with $g\alpha = u_\alpha^-$ for $\alpha \leqslant L$ and $\mathsf{elim}\ A\ \vec{u}\ a = a$ if $\vec{u}$ is an empty system.

Let us assume to have two systems M, N and $L = M, N$ is the union of these two systems. If we have $a : A$ and t_α with $u_\alpha^- t_\alpha = a\alpha$ for $\alpha \leqslant M$, then it is possible to find $v_\beta : T_\beta$ for $\beta \leqslant N$ with $q_\beta : \mathsf{Id}\ A\beta\ a\beta\ v_\beta$ such that $q_\beta\alpha_1$ is the constant path $\langle i\rangle a\beta\alpha_1$ whenever $\beta\alpha_1 = \alpha\beta_1$. We can then consider

$$a' = \mathsf{comp}\ A\ a\ [\beta \mapsto q_\beta]$$

which satisfies $a'\alpha = a\alpha = u_\alpha^- t_\alpha$ for $\alpha \leqslant M$ and $a'\beta = u_\beta^- v_\beta$ for $\beta \leqslant N$.

This defines an operation

$$(a', \vec{v}) = \mathsf{extend}\ a\ \vec{t}\,[\alpha \mapsto u_\alpha]\ [\beta \mapsto u_\beta]$$

which satisfies $a'\alpha = a\alpha$ for $\alpha \leqslant M$ and $a'\beta = u_\beta^- v_\beta$ for $\beta \leqslant N$.

The element $(a', \vec{t}, \vec{v})$ is then an element of $A\vec{u}$.

Composition for glueing

We have two systems on Γ. One system L for defining $A\vec{u} = B$ so that $\vec{u}$ is a system of isomorphisms $[\alpha \mapsto u_\alpha]$ for $\alpha \leqslant L$. One system for $b : B$ of the form $[\beta \mapsto b_\beta]$ for $\beta \leqslant J$. We write $g = \mathsf{elim}\ A\ \vec{u} : B \to A$ and define

$$c = \mathsf{comp}^i\ A\ (g\ b)\ (g\ \vec{b})$$

and, for $\alpha \leqslant L$

$$d_\alpha = \mathsf{comp}^i\ T_\alpha\ b\alpha\ \vec{b}\alpha : T_\alpha$$

We have an equality $p_\alpha = \mathsf{pres}\ g\alpha\ \vec{b}\alpha : \mathsf{Id}\ A\alpha\ c\alpha\ d_\alpha$ for $\alpha \leqslant L$ and we define

$$\mathsf{comp}^i B\ b\ [\beta \mapsto b_\beta] = ([\alpha \mapsto d_\alpha], \mathsf{comp}\ A\ c\ [\alpha \mapsto p_\alpha])$$

Transport for glueing

We have one system of isomorphisms $u_\alpha : \mathsf{Iso}(A\alpha, T_\alpha)$ for $\alpha \leqslant L$. We write $A\vec{u} = B$ and define g to be the map $\mathsf{elim}\ A\ \vec{u} : B \to A$. We have $g\alpha = u_\alpha^- : T_\alpha \to A\alpha$ if $\alpha \leqslant L$. Given b_0 in $B(i0)$, we want to define

$$\mathsf{transp}^i\ B\ b_0 : B(i1)$$

We separate $L = L', L_0, L_1$ in 3 parts: $\alpha \mapsto u_\alpha$ with α independent of i, $(i0)\beta \mapsto u_{\beta(i0)}$ and $(i1)\gamma \mapsto u_{\gamma(i1)}$. We have

$$\vec{u}(i1) = [\alpha \mapsto u_\alpha(i1)], [\gamma \mapsto u_{\gamma(i1)}]$$

We consider $a_1 = \mathsf{transp}^i\ A\ (g(i0)\ b_0) : A(i1)$ and $t_\alpha = \mathsf{transp}^i\ T\alpha\ b_0\alpha : T_\alpha(i1)$. We have for each α

$$p_\alpha = \mathsf{pres}^i\ g\alpha\ b_0\alpha : \mathsf{Id}\ A(i1)\alpha\ a_1\alpha\ (g\alpha(i1)\ t_\alpha)$$

so that we can form $a_1' = \mathsf{comp}\ A(i1)\ a_1\ \vec{p}$ which satisfies

$$a_1'\alpha = u_\alpha^-(i1)\ t_\alpha : A(i1)\alpha$$

We can then define

$$(a_1'', \vec{v}) = \mathsf{extend}\ a_1'\ \vec{t}\ [\alpha \mapsto u_\alpha(i1)]\ [\gamma \mapsto u_{\gamma(i1)}]$$

which satisties $a_1''\alpha = a_1'\alpha$ and

$$\mathsf{transp}^i\ B\ b_0 : B(i1) = (a_1'', [\alpha \mapsto t_\alpha], [\gamma \mapsto v_\gamma])$$

Composition of types

$$\frac{\Gamma \vdash A \qquad \Gamma\alpha \vdash P_\alpha : \mathsf{ID}\ A\alpha\ T_\alpha}{\Gamma \vdash A\vec{P}}$$

$$\frac{\Gamma \vdash a : A \qquad \Gamma\alpha \vdash P_\alpha : \mathsf{ID}\ A\alpha\ T_\alpha \qquad \Gamma\alpha \vdash P_\alpha^- t_\alpha = a\alpha : A\alpha}{\Gamma \vdash (\vec{t}, a) : A\vec{P}}$$

$$\frac{\Gamma \vdash A \qquad \Gamma\alpha \vdash P_\alpha : \mathsf{ID}\ A\alpha\ T_\alpha}{\Gamma \vdash \mathsf{elim}\ A\ \vec{P} : A\vec{P} \to A}$$

Composition for types composition

Given $P : \mathsf{ID}\ A\ T$ and a system $i : \mathbb{I} \vdash t_\alpha$ compatible with $t : T$ we can consider $v_0 = \mathsf{comp}^i\ A\ (P^- t)\ P^- \vec{t}$ and $v_1 = P^-(\mathsf{comp}\ i\ T\ t\ \vec{t})$, we define

$$p = \mathsf{pres}\ P\ t\ \vec{t} : \mathsf{Id}\ A\ v_0\ v_1$$

such that p_α is the constant path $\langle i\rangle(P^- t_\alpha(i1))$

This operation is defined in such a way that p is the constant path $\langle j\rangle \mathsf{comp}^i\ A\ t\ \vec{t}$ if P is constant.

We define $u = \mathsf{transp}^k\ P(j \wedge 1-k)\ t$ so that $u : Pj$ and $u(j0) = t : T$ and $u(j1) = P^- t : A$. Similarly we introduce $u_\alpha = \mathsf{transp}^k\ P\alpha(j \wedge 1-k)\ t_\alpha$. We can then consider $w = \mathsf{comp}^i\ P\ u\ \vec{u}$ which is such that $w(j0) = \mathsf{comp}^i\ T\ t\ \vec{t}$ and $w(j1) = v_0$. We define then $p = \langle j\rangle \mathsf{transp}^k\ P(j \vee k)\ w$.

We have two systems on Γ. One system L for defining $A\vec{P} = B$ so that $\vec{P}$ is a system of type equalities $[\alpha \mapsto P_\alpha]$ for $\alpha \leqslant L$. One system for $b : B$ of the form $[\beta \mapsto b_\beta]$ for $\beta \leqslant J$. We write $g = \mathsf{elim}\ A\ \vec{P} : B \to A$ and define

$$c = \mathsf{comp}^i\ A\ (g\ b)\ (g\ \vec{b})$$

and, for $\alpha \leqslant L$

$$d_\alpha = \mathsf{comp}^i\ T_\alpha\ b\alpha\ \vec{b}\alpha : T_\alpha$$

We have an equality $p_\alpha = \mathsf{pres}\ P\alpha\ b\alpha\ \vec{b}\alpha : \mathsf{Id}\ A\alpha\ c\alpha\ d_\alpha$ for $\alpha \leqslant L$ and we define

$$\mathsf{comp}^i B\ b\ [\beta \mapsto b_\beta] = ([\alpha \mapsto d_\alpha], \mathsf{comp}\ A\ c\ [\alpha \mapsto p_\alpha])$$

Transport for type composition

We have one system of equalities $P_\alpha : \mathsf{ID}\ A\alpha\ T_\alpha$ for $\alpha \leqslant L$. We write $A\vec{P} = B$ and define g to be the map $\mathsf{elim}\ A\ \vec{P} : B \to A$. We have $g\alpha = P_\alpha^- : T_\alpha \to A\alpha$ if $\alpha \leqslant L$. Given b_0 in $B(i0)$, we want to define

$$\mathsf{transp}^i\ B\ b_0 : B(i1)$$

We separate $L = L', L_0, L_1$ in 3 parts: $\alpha \mapsto u_\alpha$ with α independent of i, $(i0)\beta \mapsto u_{\beta(i0)}$ and $(i1)\gamma \mapsto u_{\gamma(i1)}$. We have

$$\vec{P}(i1) = [\alpha \mapsto P_\alpha(i1)], [\gamma \mapsto P_{\gamma(i1)}]$$

We consider $a_1 = \mathsf{transp}^i\ A\ (g(i0)\ b_0) : A(i1)$ and $t_\alpha = \mathsf{transp}^i\ T\alpha\ b_0\alpha : T_\alpha(i1)$. We have for each α

$$p_\alpha = \mathsf{pres}^i\ g\alpha\ b_0\alpha : \mathsf{Id}\ A(i1)\alpha\ a_1\alpha\ (g\alpha(i1)\ t_\alpha)$$

so that we can form $a_1' = \mathsf{comp}\ A(i1)\ a_1\ \vec{p}$ which satisfies

$$a_1'\alpha = u_\alpha^-(i1)\ t_\alpha : A(i1)\alpha$$

We can then define

$$(a_1'', \vec{v}) = \mathsf{extend}\ a_1'\ \vec{t}\ [\alpha \mapsto P_\alpha(i1)]\ [\gamma \mapsto P_{\gamma(i1)}]$$

which satisties $a_1''\alpha = a_1'\alpha$ and

$$\mathsf{transp}^i\ B\ b_0 : B(i1) = (a_1'', [\alpha \mapsto t_\alpha], [\gamma \mapsto v_\gamma])$$

In general, if we have a compatible system of equality

$$[\alpha \mapsto P_\alpha]\ [\beta \mapsto P_\beta]$$

with $P_\alpha : \mathsf{ID}\ A\alpha\ T_\alpha$ and $P_\beta : \mathsf{ID}\ A\beta\ T_\beta$ we can define

$$(a', \vec{v}) = \mathsf{extend}\ a\ \vec{t}\ [\alpha \mapsto P_\alpha]\ [\beta \mapsto P_\beta]$$

satisfies $a'\alpha = a\alpha$ for $\alpha \leqslant M$ and $a'\beta = P_\beta^- v_\beta$ for $\beta \leqslant N$. Furthermore, it is such that $a' = a$ if each P_β is constant.

Similarly, if we have $P : \mathsf{ID}\ A\ T$ then P^- does not need to preserve composition for judgemental equality. However, if we have $t : T$ and system $i : \mathbb{I} \vdash t_\alpha : T\alpha$ we can consider the composition of the images $v_0 = \mathsf{comp}^i\ A\ (P^-\ t)\ (P^-\ \vec{t})$ and the image of the composition $v_1 = P^-\ (\mathsf{comp}^i\ T\ t\ \vec{t})$ and we have an equality

$$\vdash \mathsf{pres}\ u\ \vec{t} : \mathsf{Id}\ A\ v_0\ v_1$$

which satisfies $(\mathsf{pres}\ u\ \vec{t})\alpha = \langle i\rangle(u\alpha\ t_\alpha(i1))$ for $\alpha \leqslant L$ *and* is constant if P is constant.

Comment

Constants

We use the following constants

1. $\mathsf{comp}^i\ A\ a\ \vec{u}$ with $a : A$ and $i : \mathbb{I} \vdash u_\alpha : A\alpha$, defined by induction on A
2. $\mathsf{Comp}^i\ A\ a\ \vec{u}$ with $a : A(i0)$ and $i : \mathbb{I} \vdash A$ and $i : \mathbb{I} \vdash u_\alpha : A\alpha$, defined from comp
3. $\mathsf{transp}^i\ A\ a_0$ with $a_0 : A(i0)$ and $i : \mathbb{I} \vdash A$, defined by induction on A
4. $\mathsf{pres}\ u\ t\ \vec{t}$ with $u : \mathsf{Iso}(T, A)$ and $t : T$, defined using comp
5. $\mathsf{extend}\ a\ \vec{t}\ [\alpha \mapsto u_\alpha]\ [\beta \mapsto u_\beta]$ with $a\alpha = u_\alpha^- t_\alpha$, defined using that isomorphisms are equivalence

These constant commute all with substitution. For instance, if $\Gamma \vdash A$ and $\Gamma\alpha, i : \mathbb{I} \vdash u_\alpha : A\alpha$ and $\sigma : \Delta \to \Gamma$ we have

$$\Delta \vdash (\mathsf{comp}^i\ A\ a\ \vec{u})\sigma = \mathsf{comp}^j\ A\sigma\ a\sigma\ \vec{u}(\sigma, i = j) : A\sigma$$

for any j fresh for Δ.

Glueing and composition of types

The rules for glueing and composition of types are similar. However we could not unify them: if all u_α are identity functions, then $A\vec{u}$ does not have in general the same composition operation as A, while if all E_α are constant then $A\vec{E}$ and A have the same composition operations and we have $A\vec{E} = A$.

Semantics

Each context Γ is interpreted by a cubical set as in [5]. Concretely, for each finite set of symbols I, we have a set $\Gamma(I)$ and we have restriction maps $\rho \longmapsto \rho f,\ \Gamma(I) \to \Gamma(J)$ for each $f : I \to J$ satisfying $\rho 1_I = \rho$ and $(\rho f)g = \rho(fg)$. A type $\Gamma \vdash A$ is interpreted by a family of sets $A\rho$ for each I and ρ in $\Gamma(I)$ and restriction maps $u \longmapsto uf,\ A\rho \to A\rho f$ satisfying $u1_I = u$ and $(uf)g = u(fg)$. An element $\Gamma \vdash a : A$ is interpreted by a family of element $a\rho$ in $A\rho$ such that $(a\rho)f = a(\rho f)$.

Furthermore this should have *composition* and *transport* operations. For composition, we should have an operation $u|_i\vec{u}$ in $A\rho$ for u in $A\rho$ and u_α in $A\rho\alpha\iota_i$ is a compatible family such that $u\alpha = u_\alpha(i0)$. This operation should be regular and uniform. The *regularity* is that $u|_i(\vec{u}, \alpha \mapsto u\alpha) = u|_i\vec{u}$. The uniformity is that $(u|_i\vec{u})f = uf|_j\vec{u}(f, i = j)$ if $f : I \to J$ and j not in J.

For transport, we should have an operation $\mathsf{comp}^j(u)$ in $A\rho(j1)$ if j in J and u in $A\rho(j0)$. This operation should be *regular*: if ρ is independent of j, i.e. $\rho = \rho(j0)\iota_j$, then $\mathsf{comp}^j(u) = u$ and *uniform*: $\mathsf{comp}^j(A\rho, u)f = \mathsf{comp}^k(A\rho(f, j = k), uf)$ if $f : I - j \to J$ and k is not in J.

References

[1] M.Bezem, Th. Coquand and S. Huber. A model of type theory in cubical sets. Preprint, 2013.

[2] B. van der Berg and R. Garner. Topological and simplicial models of identity types. ACM Transactions on Computational Logic (TOCL), Volume 13, Number 1 (2012).

[3] R. Brown, P. J. Higgins and R. Sivera. *Nonabelian Algebraic Topology: Filtered spaces, crossed complexes, cubical homotopy groupoids.* volume 15 of EMS Monographs in Mathematics , European Mathematical Society, 2011.

[4] H. Cartan. Sur le foncteur $Hom(X, Y)$ en théorie simpliciale. *Séminaire Henri Cartan*, tome 9 (1956-1957), p. 1-12

[5] Th. Coquand. Course notes on cubical type theory.

[6] D. Kan. Abstract Homotopy I. Proc. Nat. Acad. Sci. U.S.A., 41 (1955), p. 1092-1096.

[7] J.C. Moore. Lecture Notes, http://faculty.tcu.edu/gfriedman/notes/, Princeton 1956 p. 1A-8.

[8] A. M. Pitts. An Equivalent Presentation of the Bezem-Coquand-Huber Category of Cubical Sets. Manuscript, 17 September 2013.

[9] R. Williamson. Combinatorial homotopy theory. Preprint, 2012.