Simplicial set model of HIT cones

Vladimir Voevodsky

Started March 21, 2013. Work in progress.

1 Semantics of identity types

We consider identity types (which we denote "paths") which are introduced by the following group of inference rules:

$$[\mathbf{r1}]\frac{\begin{array}{c}\Gamma \vdash T\ type \\ \Gamma \vdash o : T\end{array} \quad \Gamma \vdash o' : T}{\Gamma \vdash paths(o, o')\ type} \tag{1}$$

$$[\mathbf{r2}]\frac{\begin{array}{c}\Gamma \vdash T\ type \\ \Gamma \vdash o : T\end{array}}{\Gamma \vdash idpath(o) : paths(o, o)} \tag{2}$$

$$[\mathbf{r3}]\frac{\begin{array}{ccc}\Gamma \vdash T\ type & \Gamma, x : T, e : paths(o, x) \vdash P\ type & \Gamma \vdash o' : T \\ \Gamma \vdash o : T & \Gamma \vdash s0 : P[o/x, (idpath\ o)/e] & \Gamma \vdash e' : paths(o, o')\end{array}}{\Gamma \vdash paths\_rect(o, T, x.e.P, s0, o', x') : P[o'/x, e'/e]} \tag{3}$$

$$[\mathbf{r4}]\frac{\begin{array}{cc}\Gamma \vdash T\ type & \Gamma, x : T, e : paths(o, x) \vdash P\ type \\ \Gamma \vdash o : T & \Gamma \vdash s0 : P[o/x, (idpath\ o)/e]\end{array}}{\Gamma \vdash (paths\_rect(o, T, x.e.P, s0, o, idpath(o)) \stackrel{d}{=} (so\ o)} \tag{4}$$

If $\mathcal{C}$ is a locally Cartesian closed category and $p : \tilde{\mathcal{U}} \to \mathcal{U}$ is a universe in $\mathcal{C}$.

then a paths-structure on the C-system $CC(\mathcal{C}, p)$ can be introduced as follows.

Let us denote by B_n the set of objects of length n in $CC(\mathcal{C}, p)$ and by $\widetilde{B}_n$ the set of sections of the canonical projections $X \to ft(X)$ where $X \in B_n$. For $\Gamma \in B_n$ let $B(\Gamma)$ be the set of elements Γ' such that $ft(\Gamma') = \Gamma$ and $\widetilde{B}(\Gamma')$ the set of elements s such that $\partial(s) = \Gamma'$.

By construction of $CC(\mathcal{C}, p)$ there is an object $[\Gamma]$ of $\mathcal{C}$ corresponding to each $\Gamma \in B_n$ and one has natural bijections:

$$[\mathbf{b1}]B(\Gamma) = Hom_{\mathcal{C}}([\Gamma], \mathcal{U}) \tag{5}$$

$$[\mathbf{b2}]\widetilde{B}(\Gamma') = Hom_{\mathcal{U}}([\Gamma'], \tilde{\mathcal{U}}) \tag{6}$$

The rule (1) corresponds to specifying for all s an operation of the form

$$paths : \widetilde{B}(\Gamma)_{\partial} \times_{\partial} \widetilde{B}(\Gamma) \to B(\Gamma)$$

From the bijections (5),(6) we conclude that such an operation can be obtained from a morphism

$$Paths : \tilde{\mathcal{U}} \times_{\mathcal{U}} \tilde{\mathcal{U}} \to \mathcal{U}$$

Rule (2) corresponds to specifying for all Γ an operation of the form

$$idpath : \widetilde{B}(\Gamma) \to \widetilde{B}(\Gamma)$$

such that $\partial(idpath(s)) = paths(s,s)$. From the bijections (5),(6) we conclude that such an operation can be obtained from a morphism

$$Idpath : \tilde{\mathcal{U}} \to \tilde{\mathcal{U}}$$

such that $p \circ Idpath = Paths \circ \Delta$ where $\Delta : \tilde{\mathcal{U}} \to \tilde{\mathcal{U}} \times_{\mathcal{U}} \tilde{\mathcal{U}}$ is the diagonal.

Let us consider now the third rule (3). As an operation it has five arguments. The first argument is as before an element s of $\widetilde{B}(\Gamma)$. The second argument is an element of $B(\Gamma')$ where Γ' is expressible through s (given $paths$). To find the expression for Γ' consider its derivation tree. We have

$$\frac{\Gamma, x : T \vdash o : T \quad \Gamma, x : T \vdash x : T}{\Gamma, x : T \vdash paths\ o\ x\ type}$$

We have $(\Gamma, x : T \vdash o : T) = \widetilde{T}(\partial(s), s)$ and $(\Gamma, x : T \vdash x : T) = \delta(\partial(s))$. Therefore the second argument of rule (3) is an element Γ' of $B(paths(\widetilde{T}(\partial(s), s), \delta(\partial(s))))$. The third argument is an element s' of $\widetilde{B}(\Gamma)$ with the condition that $\partial(s')$ is expressible through s and Γ' (given $paths$ and $idpath$). To find the exact form of the expression consider the derivation tree for the sentence $(\Gamma \vdash P[o/x, idpath(o)/e]\ type)$. We have:

$$\frac{\Gamma \vdash o : T \quad \Gamma, x : T, e : paths\ o\ x \vdash P\ type}{\Gamma, e : paths\ o\ o \vdash P[o/x]\ type}$$

$$\frac{\Gamma \vdash idpath(o) : paths\ o\ o \quad \Gamma, e : paths\ o\ o \vdash P[o/x]\ type}{\Gamma \vdash P[o/x, idpath(o)/e]\ type}$$

Therefore we have

$$\partial(s') = S(idpath(s), S(s, \Gamma'))$$

Finally the result of the operation defined by (3) is an element r of $\widetilde{B}(\Gamma)$ with the condition that

$$\partial(r) = \prod(\prod(\Gamma'))$$

Let us now describe the domain of definition of the rule (3) as the set of morphisms from $[\Gamma]$ to some object A of $\mathcal{C}$. The first argument is specified by a morphism $s : [\Gamma] \to A_1$. As we know $A_1 = \tilde{\mathcal{U}}$. The second argument belongs to the set $B(paths(\delta(\partial(s)), \widetilde{T}(\partial(s), s)))$ which varies over $\widetilde{B}(\Gamma)$. Therefore the representing object is an object over $\tilde{\mathcal{U}}$.

To describe in the same way the second argument let us compute the object which represents $B(paths(\delta(\partial(s)), \widetilde{T}(\partial(s), s)))$ for a given s. Since $s : [\Gamma] \to \tilde{\mathcal{U}}$ the will be object over $\tilde{\mathcal{U}}$. By construction

$$B(paths(s_1, s_2)) = Hom_{\mathcal{C}}([\Gamma]_{Paths \circ (s_1, s_2)} \times_{\mathcal{U}} \tilde{\mathcal{U}}, \mathcal{U})$$

1 Semantics of cones

Cones of functions $f : X \to Y$ together with the associated constructors, eliminator and the computation rule provide an important example of higher inductive types. In this note we show how to extend the univalent simplicial set model to this group of structures.

Formally, the cones structure in an intensional type system is defined by the following inference rules:

$$\frac{\Gamma \vdash f : X \to Y}{\Gamma \vdash \mathit{cone}(f)\ \mathit{type}}$$

$$\frac{\Gamma \vdash f : X \to Y}{\Gamma \vdash \mathit{conec}0(f) : \mathit{cone}(f)}$$

$$\frac{\Gamma \vdash f : X \to Y}{\Gamma \vdash \mathit{conec}1(f) : Y \to \mathit{cone}(f)}$$

$$\frac{\Gamma \vdash f : X \to Y}{\Gamma \vdash \mathit{conec}2(f) : \mathit{forall}\ x : X, \mathit{paths}\ \mathit{conec}0(f)\ (\mathit{conec}1(f)\ (f\ x))}$$

$$\frac{\begin{array}{l}\Gamma \vdash f : X \to Y \\ \Gamma, c : \mathit{cone}(f) \vdash P\ \mathit{type} \\ \Gamma \vdash c0 : P[\mathit{conec}0(f)/c] \\ \Gamma \vdash c1 : \mathit{forall}\ y : Y, P[(\mathit{conec}1(f)\ y)/c] \\ \Gamma \vdash c2 : \mathit{forall}\ x : X, \mathit{paths}\ (\mathit{tpair}\ \mathit{conec}0(f)\ c0)\ (\mathit{tpair}\ (\mathit{conec}1(f)\ (f\ x))(c1\ (f\ x)))\end{array}}{\Gamma \vdash \mathit{cone\_rect}(f, c.P, c0, c1, c2) : \mathit{forall}\ c : \mathit{cone}(f), P}$$

$$\frac{\begin{array}{l}\Gamma \vdash f : X \to Y \\ \Gamma, c : \mathit{cone}(f) \vdash P\ \mathit{type} \\ \Gamma \vdash c0 : P[\mathit{conec}0(f)/c] \\ \Gamma \vdash c1 : \mathit{forall}\ y : Y, P[(\mathit{conec}1(f)\ y)/c] \\ \Gamma \vdash c2 : \mathit{forall}\ x : X, \mathit{paths}\ (\mathit{tpair}\ \mathit{conec}0(f)\ c0)\ (\mathit{tpair}\ (\mathit{conec}1(f)\ (f\ x))(c1\ (f\ x)))\end{array}}{\Gamma \vdash (\mathit{cone\_rect}(f, c.P, c0, c1, c2)\ \mathit{conec}0(f)) \stackrel{d}{=} c0}$$

$$\frac{\begin{array}{l}\Gamma \vdash f : X \to Y \\ \Gamma, c : \mathit{cone}(f) \vdash P\ \mathit{type} \\ \Gamma \vdash c0 : P[\mathit{conec}0(f)/c] \\ \Gamma \vdash c1 : \mathit{forall}\ y : Y, P[(\mathit{conec}1(f)\ y)/c] \\ \Gamma \vdash c2 : \mathit{forall}\ x : X, \mathit{paths}\ (\mathit{tpair}\ \mathit{conec}0(f)\ c0)\ (\mathit{tpair}\ (\mathit{conec}1(f)\ (f\ x))(c1\ (f\ x))) \\ \Gamma \vdash y0 : Y\end{array}}{\Gamma \vdash (\mathit{cone\_rect}(f, c.P, c0, c1, c2)\ (\mathit{conec}1(f)\ y0)) \stackrel{d}{=} (c1\ y0)}$$

$$\frac{\begin{array}{l}\Gamma \vdash f : X \to Y \\ \Gamma, c : cone(f) \vdash P\ type \\ \Gamma \vdash c0 : P[conec0(f)/c] \\ \Gamma \vdash c1 : forall\ y : Y, P[(conec1(f)\ y)/c] \\ \Gamma \vdash c2 : forall\ x : X, paths\ (tpair\ conec0(f)\ c0)\ (tpair\ (conec1(f)\ (f\ x))(c1\ (f\ x))) \\ \Gamma \vdash x0 : X\end{array}}{\Gamma \vdash (maponpaths\ tcone\_rect\ (conec2(f)\ x0) \stackrel{d}{=} (c2\ x0)}$$

Where *paths* is the identity type of the type system, *tpair* is the pair formation constructor of the dependent sums, $tcone_rect$ is the function

$$tcone_rect := fun\ c : cone(f) \Rightarrow tpair\ c\ (cone\_rect(f, c.P, c0, c1, c2)\ c)$$

and *maponpaths* is the function on paths obtained in the usual way using the eliminator for the identity types.