A very short note on homotopy λ-calculus

Vladimir Voevodsky

Started September 23, 2006

Here is the general picture as I understand it at the moment. Let us consider the type system TS which is generated by the following rules:

1. $$\frac{\vdash}{\vdash U_i : U_{i+1}}$$
   for all $i = -1, 0, 1, \ldots$
2. $$\frac{\Gamma \vdash T : U_i}{\Gamma \vdash T : U_{i+1}}$$
3. $$\frac{\Gamma \vdash T : U_i}{\Gamma \vdash T : Type}$$
4. The usual dependent $\prod$-rules (inside each U_n)
5. The usual dependent $\sum$-rules with strong elimination (inside each U_n)[1]

Let CC be the contexts category of TS[2]. By a model of TS with values in a category D I mean a functor $CC \to D$ which "preserves the relevant structures".

The main observation is that there is a canonical model M of TS with values in the usual homotopy category H provided that we consider homotopy types based on a sufficiently large universe of sets. To define this model one starts with a not-so-canonical model N of TS with values in the category of spaces (actually simplicial sets, but I will use spaces as a more intuitive model for homotopy types) and then sets M to be the composition of N with the projection $Spc \to H$ is M. Here are the defining properties of N.

1. By definition N takes a context Γ to a space $N(\Gamma)$.
2. A sequent of the form $\Gamma \vdash T : Type$ (where T is an expression) defines a morphism
   $$(\Gamma, x : T) \to \Gamma$$
   in CC. Morphisms of this type go to *fibrations*
   $$N(\Gamma; T) : N(\Gamma, x : T) \to N(\Gamma),$$

[1] We may also consider systems TS_n where U_i are only defines for $i \leq n$. E.g. TS_{-2} will be empty, TS_{-1} will have one generating type U_{-1} with no terms. It looks similar to the usual λ-calculus with one generating type. Starting with TS_0 there are real dependencies.

[2] Similarly we may consider CC_n for finite n. The category CC_{-2} is empty and CC_{-1} is probably the free Cartesian closed category generated by one object U_{-1}.

3. A sequent of the form $\Gamma \vdash t : T$ (where T and t are expressions) defines a morphism

$$\Gamma \to (\Gamma, x : T)$$

in CC. Morphisms of this type go to *sections*

$$N(\Gamma; T, t) : N(\Gamma) \to N(\Gamma, x : T)$$

of $N(\Gamma; T)$.

Given $\Gamma \vdash P : Type$ and $\Gamma, x : P \vdash Q : Type$ we can form $\Gamma \vdash \prod x : P.Q$ and $\Gamma \vdash \sum x : P.Q$. On the model level our data defines two fibrations

$$N(\Gamma, x : P, y : Q) \xrightarrow{q} N(\Gamma, x : P) \xrightarrow{p} N(\Gamma)$$

The fibration

$$N(\Gamma, z : \prod x : P.Q) \to N(\Gamma)$$

is the "$p_*(q)$". Its fiber over $x \in N(\Gamma)$ is the space of sections (continuous ones!) of the fiber of q over x.

The fibration

$$N(\Gamma, z : \sum x : P.Q) \to N(\Gamma)$$

is the "$p_!(q)$". It is simply the composition of p and q. The meaning of term constructors associated with $\sum$ and $\prod$ is the obvious one. If we took a model with values in *Sets* where all maps are fibrations we would get the usual rules for interpretation of $\sum$ and $\prod$ but formulated in a slightly unusual way.

The rigorous description of the value of N on U_n's is complicated. Up to homotopy equivalence, the space $N(U_n)$ is the nerve of the n-groupoid of $(n-1)$-groupoids in the ZF with $n-2$ universes (see below for the explicit form in the case $n \leq 1$). Since n-groupoids are the same as n-homotopy types one can also say it in a purely homotopy-theoretic way.

In particular,

$$M(U_{-1}) = \emptyset$$

and

$$M(U_0) = \{0, 1\}.$$

We further have

$$M(U_1) = \coprod_{n \geq 0} BS_n$$

where BS_n is the classifying space of the permutation group on n elements (BS_0 is empty, BS_1 is one point and BS_2 is homotopy equivalent to $\mathbf{R}P^\infty = B\mathbf{Z}/2$). In particular, $\pi_0(M(U_1)) = \mathbf{N}$ and one uses U_1 to define the type of natural numbers in $H\lambda$. As far as I understand at the moment $M(U_2)$ is $\coprod_{X \in u_2} BAut(X)$ where u_2 is the set of equivalence classes of all groupoids with sets of morphisms and objects being ZF-sets. Starting with U_2 one needs ZF with universes in order for the model to be defined. The model of U_3 is the nerve of the 3-groupoid of 2-groupoids in ZF with one universe.

The model of U_n has a natural filtration by subspaces $U_{n,k}$, $k = 0, \ldots, n$ where $U_{n,k}$ is (the nerve of) the k-groupoid of $(k-1)$-groupoids in the universe U_n. In particular $U_{n,1}$ is the (nerve of) the usual groupoid of sets in U_n and their isomorphisms. We define a (-1)-groupoid as a set where any two elements are equal i.e. one of the two sets $\emptyset$ and pt. Hence for any $n \geq 0$ the model of $U_{n,0}$ is the two point set $\{0,1\} = \{true, false\}$.

$$
\begin{tikzcd}
U_{0,0} \arrow[r, equal] & U_{1,0} \arrow[r, equal] \arrow[d] & U_{2,0} \arrow[r, equal] \arrow[d] & U_{3,0} \arrow[r, equal] \arrow[d] & \ldots \\
& U_{1,1} \arrow[r] & U_{2,1} \arrow[r] \arrow[d] & U_{3,1} \arrow[r] \arrow[d] & \ldots \\
& & U_{2,2} \arrow[r] & U_{3,2} \arrow[r] \arrow[d] & \ldots \\
& & & U_{3,3} \arrow[r] & \ldots
\end{tikzcd}
$$

All the arrows are inclusions with the image being a disjoint union of some of the connected components of the target and the usual arguments a-la Russell's paradox imply that except for the ones marked as equalities the arrows are proper inclusions e.g. $U_{2,1}$ (which is responsible for sets in U_2) is strictly larger than $U_{1,1}$ (which is responsible for sets in U_1) etc.

The model M can be used to define an hierarchy of equality types $Eq_k(T; t1, t2)$ on TS. Given a valid type expression $T : U_n$ and two term expressions $t1, t2 : T$ we get on the level of models a space $X = M(T)$ (up to homotopy) and two points $x1, x2 \in X$. The model of $Eq_k(T; t1, t2)$ for $k \geq n$ is (homotopy equivalent to) the space $P(X; x1, x2)$ of paths from $X1$ to $x2$ in X. The model of $Eq_0(T; t1, t2)$ is a truth value (i.e. is empty or contractible) which is *true* iff $x1$, $x2$ belong to the same connected component of X. In a sense Eq_0 is the Leibnitz equality type while Eq_k for $k \geq n$ is the "full" equality type. The Eq_k's are concrete (and complicated) type expressions in TS (at least for $T : U_n$ where n is fixed).

So defined, the equality types lack many of the properties which hold on the model level. The *homotopy λ-calculus* $H\lambda$ is a hypothetical extension of TS by means of additional rules which make the behavior of Eq_k's more "natural". In particular, on the level of $H\lambda$, U_{-1} becomes the empty type, U_0 becomes $Prop$ and U_1 becomes the type of finite sets which is used in the usual way to define the type of natural numbers. By construction $H\lambda$ also comes with a canonical model in the homotopy category.

Originally, I was considering a different approach to $H\lambda$ where the equality types where introduced as "primitives" along with $\sum$ and $\prod$ and the universes where "defined" but it seems to me now that it is nicer to start with $\sum$, $\prod$ and universes and define the equality types later.

The advantage of $H\lambda$ and its homotopy-theoretic model over the less sophisticated type systems is that it better reflects the way mathematicians envision "types" corresponding to mathematical structures. For example if we fix the size of the universe n and write in the usual way the type expression for, say, the type $Gr(U_n)$ of groups in U_n then the model of $Gr(U_n)$ will be (the nerve of) the groupoid of groups in the universe U_n and their isomorphisms. Similarly, if we write down the definition of a category in a proper way then the model of $Cat(U_n)$ will be (the nerve of) the

2-groupoid of categories in U_n, their equivalences and natural isomorphisms between equivalences. Moreover, any construction on categories described in the language of $H\lambda$ is automatically "invariant" under equivalences of categories. E.g. any function we can describe in $H\lambda$ from $Cat(U_n)$ to $Gr(U_n)$ will on the model level correspond to a construction which produces a group from a category which maps equivalences between categories to isomorphisms between the corresponding groups. In the usual type systems we can do something like that for types of "level 1" (see below) i.e. sets with structures but not for higher levels (e.g. categories).

At the moment much of what I said above is at the level of conjectures. Even the definition of the model of TS in the homotopy category is non-trivial. Similarly, the definition of equality types in terms of universes is rather involved and I am not sure which of the properties of these types have to be imposed so that the rest will follow.

In the rest of the text I will try to give one possible definition of the equality types in TS. One proceeds in the following steps:

Define the contractibility on the level of TS. Set

$$true = (U_{-1} \to U_{-1}) : U_0$$

$$false = U_{-1} : U_0.$$

For $T, T' : U_0$ set

$$Equiv(T, T') = (T \to T') \times (T' \to T).$$

For $T : U_n$ set

$$Contr(T) = \prod F : U_n \to U_0.Equiv(F(T), F(true))$$

then $M(Contr(T)) \neq \emptyset$ iff $M(T)$ belongs to the same connected component of $M(U_n)$ as $M(true)$ i.e. if $M(T)$ is a contractible space. In that case $M(Contr(T))$ is itself contractible.

Define representable functors on the level of TS. Suppose $T : U_n$ is a type (expression). I want to think of its model $X = M(T)$ as of the nerve of some n-groupoid in U_n. The members of T correspond to objects. For $T = U_n$ we get the groupoid of all groupoids. Functions $T \to U_n$ correspond to functors from T to the groupoid of all groupoids. Among these functors there are representable ones i.e. we have the homotopy type $Rep(T)$ which maps to $T \to U_n$. For $F : T \to U_n$ set

$$rep(F) = Contr(\sum t : T.F(T)).$$

One verifies that on the level of models $rep(F) \neq \emptyset$ iff F is representable. Set

$$Rep(T) = \sum F : T \to U_n.rep(F).$$

then the model of $Rep(T)$ is the space of representable functors on T. By abuse of notation I will write $F(t) : U_n$ instead of the formal $(\pi F)(t)$ for $F : Rep(T)$ and $t : T$.

Define $U_{n,k}$ on the level of TS. We first define type expressions $Lv_k(T) : U_0$ for $T : U_n$ which are "indicator functions" for $U_{n,k}$. Start with $k = 0$. Note that for a "representable functor"

$F : Rep(T)$ and $t : T$ the value $F(t)$ is the type of equivalences between the representing member and t. Hence, T is of level 0 iff for all F and all t the type $F(t)$ is contractible. I.e.

$$Lv_0(T) = \prod F : Rep(T). \prod t : T. Contr(F(t))$$

Similarly T is of level k iff all $F(t)$ are of level $k-1$. Hence for $k \geq 1$ we have

$$Lv_k(T) = \prod F : Rep(T). \prod t : T. Lv_{k-1}(F(t))$$

We can set now:

$$U_{n,k} = \sum T : U_n. Lv_k(T)$$

(in this numbering sets are of level 1 and usual 1-groupoids are of level 2.) On the model level we have $U_{n,k} = U_n$ for $k \geq n$. I do not know if this is provable in TS. If not add this as an axiom.

Define Eq_k, $Hfiber_k$, $isheq_k$ and Heq_k. We are going to define by common induction on k starting with $k = 0$ the following type expressions:

1. $Eq_k(T; t1, t2) : U_n$ for $T : U_n$ and $t1, t2 : T$ – for $k \geq n$ the model of Eq_k will be $P(X; x1, x2)$ where X is the model of T and $x1, x2$ the models of $t1, t2$,
2. $Hfiber(f, t) : U_n$ for $f : T' \to T$, $t : T$ and $T, T' \in U_n$ – for $k \geq n$ the model of $Hfiber(f, t)$ is the homotopy fiber of the model of f over the model of t,
3. $isheq_k(f) : U_0$ for f as above – for $k \geq n$ the model of f is a homotopy equivalence iff the model of $isheq_k(f)$ is $true$,
4. $Heq_k(T', T) : U_n$ for $T', T : U_n$ – for $k \geq n$ the model of $Heq_k(T', T)$ is the type of homotopy equivalences from T' to T.

We start with:

$$Eq_0(T; t1, t2) = true$$

and proceed as follows:

$$Hfiber_k(f, t) = \sum t' : T'. Eq_k(t', f(t))$$

$$isheq_k(f) = \prod t : T. Contr(Hfiber_k(f, t))$$

$$Heq_k(T', T) = \sum f : T' \to T. isheq_k(f)$$

$$Eq_{k+1}(T; t1, t2) = \prod F : Rep(T). Heq_k(F(t1), F(t2))$$

in the last expression I write $F(t)$ for $F : Rep(T)$ instead of the correct but long $(\pi F)(t)$. Let us "prove" that these expressions do indeed have the required models for $k \geq n$. We want to show a more detailed thing namely that the models are of the required form for $T, T' : U_{n,k}$ no matter what k is.

We proceed by induction on k. For $k = 0$ we get:

$$Eq_0(T; t1, t2) = true$$

$$Hfiber_0(f, t) = \sum t' : T'. Eq_0(t', f(t)) = T'$$

$$isheq_0(f) = \prod t : T.Contr(Hfiber_0(f,t)) = \prod t : T.Contr(T') = T \to Contr(T')$$

since $T, T' : U_{n,0} = U_0$ (we are on the model level) it means that T and T' are truth values. Since we are given $f : T' \to T$ we know that $T' \Rightarrow T$. We also know that for $T' : U_0$ one has $Contr(T') = T'$. A map of truth values $T' \to T$ is an equivalence iff there exists a map $T \to T'$. OK.

Assume all is well for k and consider $k+1$. First let's check that the model of

$$Eq_{k+1}(T; t1, t2) = \prod F : Rep(T).Heq_k(F(t1), F(t2))$$

is indeed the space of paths or, from the point of view of higher groupoids that it is the k-groupoid of equivalences between objects $t1$, $t2$ of T.

For $T : U_{n,k+1}$ and $F : Rep(T)$ we have by definition $F(t) : U_{n,k}$. By induction the model of $Heq_k(F(t1), F(t2))$ is the space of homotopy equivalences from $F(t1)$ to $F(t2)$ i.e. the groupoid of equivalences between k-groupoids $F(t1)$, $F(t2)$. Let t be the object which represents F. Then $F(t1)$ (resp. $F(t2)$) is the groupoid of equivalences from t to $t1$ (resp. $t2$). If $t1$ is not equivalent to $t2$ then F represented by $t1$, $F(t2)$ is empty while $F(t1)$ is not and the product is empty. If $t1 \cong t2$ then the product will contain only one term different from the point and it will be exactly the groupoid of equivalences from ...

$$Heq_0(T', T) = \sum f : T' \to T.isheq_0(f)$$

Consider the evaluation map restricted to representable functors

$$rev : T \to ((T \to U_n) \to U_n) \to (Rep(T) \to U_n)$$

This map should be a full embedding. Hence we would expect that for $t1, t2 : T$ one has

$$Eq(T; t1, t2) = Eq((Rep(T) \to U_n), rev(t1), rev(t2))$$

On the other hand for $F, G : T \to T'$ one should have

$$Eq(T \to T'; F, G) = \prod t : T.Eq(T'; F(t), G(t))$$

(this is kind of functional extensionality of equality). Hence

$$[\textbf{eqform}]Eq(T; t1, t2) = \prod X : Rep(T).Eq(U_n; rev(t1)(X), rev(t2)(X)). \quad (1)$$

Thus we have reduced the problem of defining $Eq(T; t1, t2)$ for $T : U_n$ to the problem of defining $Eq(U_n; X, Y)$ for $X, Y : U_n$.

Let us now proceed to the equality types. The idea is that for $T : U_{n,k}$ the types $rev(t1)(X)$ and $rev(t2)(T)$ which appear in (**??**) are in $U_{n,k-1}$ (on the model level at least). Hence we may use the induction on k in the definition of $Eq(-; -, -)$.

If $T : U_{n,0}$ then for $t1, t2 : T$ and $F : Rep(T)$ we have $Contr(F(t1))$ and $Contr(F(t2))$. Clearly for contractible $T1, T2$ one has $Eq(U_n; T1, T2) = pt$. Hence $Eq(T; t1, t2) = pt$ in this case.

In general, for $T1, T2 : U_n$ a member of $Eq(U_n; T1, T2)$ is a"homotopy eqivalence" from $T1$ to $T2$ i.e. a map $f : T1 \to T2$ such that all its homotopy fibers are contractible. The trick is to define the homotopy fiber of f over $t : T2$. From homotopy theory we have a formula:

$$hfiber(f,t) = \sum t' : T1.Eq(T2; t, f(t'))$$

I will write $f^{-1}(t)$ instead of $hfiber(f,t)$. Set

$$heq(U_n; T1, T2) = \sum f : T1 \to T2. \prod t : T2.Contr(f^{-1}(t))$$

Let $Rep_k(T)$ be the type of representable functors on T which land in $U_{n,k}$ i.e.

$$Rep_k(T) = \sum F : Rep(T). \prod t : T.Lv_k(F(t))$$

Let

$$Eq_0(T; t1, t2) = \prod F : Rep$$

$$Eq_k(T; t1, t2) = \prod F : Rep_k(T).Eq_{k-1}(U_n; F(t1), F(t2)).$$

Proceed by induction as follows:

1. For $X, Y : U_0$ set $Eq(U_0; X, Y) = Equiv(X, Y) = (X \to Y) \times (Y \to X)$
2. Assuming that $Eq(U_{n-1}; -, -)$ are defined and therefore $Eq(T; -, -)$ are defined for $T : U_n$ set:

Then $M(Rep(T))$ is the space of representable functors on T which should be equivalent to T if $T : U_n$. We may construct a map $T \to Rep(T)$ on the level of the type system as the composition of two obvious maps:

The categories CC have an additional structure. Namely, there are functors U_∞ and $\tilde{U}_\infty$ from CC to $Sets$ and a morphism $\tilde{U}_\infty \to U_\infty$ which are defined as follows. The functor U_∞ sends a context Γ to the set of all sequents of the form $\Gamma \vdash T : Type$, the functor $\tilde{U}_\infty$ sends Γ to the set of all sequents of the form $\Gamma \vdash t : T$ where $\Gamma \vdash T : Type$. The morphism is obvious. The functor U_∞ has the representable functors U_n for $n < \infty$ as its subfunctors according to the second group of rules.

Any functor $M : C \to D$ defines the inverse image functor $M^* : Funct(C, Sets) \to Funct(D, Sets)$. If F is representable by X then $M^*(F)$ is representable by $M(X)$. The functor M^* also preserves monomorphisms. Hence, any model M of T in D defines a morphism of functors $\tilde{F}_M \to F_M$ where $F_M = M^*(U_\infty)$ and similarly for $\tilde{F}_M$. The functor $F(M)$ contains $M(U_n)$ as representable subfunctors.