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)

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.

Let CC_∞ (resp. CC_n) be the contexts category of TS_∞. The category CC_{-2} is empty and CC_{-1} is probably the free Cartesian closed category generated by one object U_{-1}.

Let us consider now models of TS_∞ with values in a category D i.e. functors $CC_\infty \to D$ which "preserve the relevant structures". I claim 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. This model takes U_{-1} to $\emptyset$ and U_0 to $\{0, 1\}$. The object U_1 goes to the homotopy type $\coprod_{n \geq 0} BS_n$ where BS_n is the classifying space of the permutation group on n elements (e.g. 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, $\pi_0(M(U_1)) = \mathbf{N}$. 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 2-groupoid of 1-groupoids in ZF with one universe. In general the model of U_n is the nerve of the $(n-1)$-groupoid of $(n-2)$-groupoids in the ZF with $n-2$ universes. Since n-groupoids are the same as n-homotopy types one can also say it in a purely homotopy-theoretic way.

This model is very "incomplete" in the sense that there are many type expressions T such that $M(T)$ is non-empty while T has no terms in TS. This is of course unavoidable because of the Goedel's theorem. However, some of these incompletenesses are of a special kind. For example for any T the space $M(((T \to U_{-1}) \to U_{-1}) \to T)$ is non-empty (this is a combination of Boolean property with some weak form of the axiom of choice) while the type $((T \to U_{-1}) \to U_{-1}) \to T$ may well have no terms. There are other more sophisticated examples.

The homotopy λ-calculus $H\lambda$ is the hypothetical extension of TS_∞ which includes new rules which are necessary to prevent such "obvious" incompletenesses. There are three groups of rules which I can see right away. The Boolean rule, the axiom of choice rule and a whole group of rules which are related to the equality types. The later ones are most interesting.

Given a space X (I use the word "space" pretty much as a synonym for the "homotopy type") and two points $p, q \in X$ let $P(X; p, q)$ be the space of paths from p to q. If X is (the nerve of) a groupoid and p, q are objects then $P(X; p, q)$ is the set of isomorphisms from p to Q in X. The first non-trivial observation is that for a type expression T and two term expressions $t1, t2$ of type T it is possible to write down a type expression $eq(T; t1, t2)$ such that $M(eq(T; t1, t2)) = P(M(T); M(t1), M(t2))$. I want to call $eq(T; t1, t2)$ the equality type for $t1$, $t2$ in T. Here is one way to get it. We start with $T : U_n$ and $T1 : T$, $T2 : T$. The formula for eq will depend on n and will be defined by induction on n. Set

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

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

and 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.

For $F : T \to U_n$ set

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

On the level of models envisioned as higher groupoids the type $T \to U_n$ is the type of all functors from T to the n-groupoid of all $(n-1)$-groupoids (of bounded size) and their equivalences. Among these there are representable functors. One verifies that on the level of models $rep(F) \neq \emptyset$ iff F is representable. Set

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

i.e. $Rep(T)$ is the space of representable functors on T. 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

$$eq(T; t1, t2) = \prod X : Rep(T).eq(U_n; rev(t1)(X), rev(t2)(X)).$$

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 $X, Y : U_n$. 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.