1-Types versus Set-Based Groupoids

Nicolai Kraus

January'14

Abstract

In Homotopy Type Theory, we have (at least) two plausible notions of "groupoid":

1. 1-truncated types
2. a set of points, and for any pair of points, a set of morphisms between them, together with all the data needed to make it a groupoid (a *set-based* representation).

From a groupoid in the second representation, it seems to be fairly simple to get one in the first presentation. We do not discuss this here (it is a simple higher inductive types, see Rezk completion).

I show that we can derive the second representation from the first one (in the sense of Altenkirch) if and only if the task is trivial, i.e. for 1-types that are actually sets.

However, for a given 1-type with braided loop spaces ("weakly abelian" loop spaces), I can construct a weak form of the second representation which only includes all loop spaces instead of all path spaces.

1 Negative Results

Definition 1 (Altenkirch). Given a type A, we say that A is *presentable as a groupoid* iff there is

$$h : \|A\|_0 \times \|A\|_0 \to \mathcal{U}$$

such that

$$c : \forall (a_1\, a_2 : A).\, h(|a_1|, |a_2|) \simeq (a_1 =_A a_2).$$

Altenkirch wanted to know whether a type can always be presented as a groupoid, where the correctness criterion should probably be that the canonical HIT-construction allows to recover the original type (this makes sense at least for 1-types). Note that I don't require A to be a 1-type as this is never used in the proof of the negative result.

Capriotti observed that, if A is presentable as a groupoid, then

$$|-| : A \to \|A\|_0 \tag{1}$$

has a section. The other direction of this statement is actually wrong - a section does allow us to construct a reasonable h, but we will not get c. This would require a retraction (which would immediately imply that A is a set).

I give an improvement of this observation:

Theorem 2. *A type is presentable as a groupoid if and only if it is a set.*

The first direction ("if") holds trivially. The title of this section is "negative results" as the theorem shows Altenkirch's notion to only be attainable in the trivial case.

Proof (of the nontrivial direction of Theorem 2). I want to start with a simple observation. It probably follows from the usual "proof-rewriting-rules" and could be done in much higher generality, but for now, it's enough like this:

Observation. Given types A, B, functions $j : A \to B$ and $k : B \to \mathcal{U}$ as well as $a_1, a_2 : A$ and $p : a_1 = a_2$. Then, there are three functions $k(j(a_1)) \to k(j(a_2))$, namely the ones coming from

- transporting along p
- transporting along $\mathsf{ap}_j p$
- transporting along $\mathsf{ap}_{k \circ j} p$.

These three functions are all equal (trivial by induction on p).

Assume now that we have a type A, together with h and c as in Definition 1. I need to show that A is a set. Assume $a_1, a_2, b : A$ and $p : a_1 = a_2$. The latter proof gives us $c(|a_1|, |b|) = c(|a_2|, |b|)$. We need to analyze what exactly this means: it implies that the square

$$\begin{tikzcd}
h(|a_1|, |b|) \arrow[r, "\text{induced by } c(|a_1|,|b|)"] \arrow[d, "\text{transporting along } p"'] & a_1 = b \arrow[d, "\text{transporting along } p"] \\
h(|a_2|, |b|) \arrow[r, "\text{induced by } c(|a_2|,|b|)"'] & a_2 = b
\end{tikzcd}$$

commutes (up to homotopy). The right vertical map is just composition with p. The left vertical map can, by the above observation, be replaced by the map that is given by transporting along $|p|$ (the situation is: $A :\equiv A$, $B :\equiv \|A\|_0$, $j :\equiv |-|$, $k :\equiv \lambda x. h(x, |b|)$; and $|p|$ means $\mathsf{ap}_{|-|} p$). However, transporting along $|p|$ is equal to transporting along $|q|$ for any other $q : a_1 = a_2$, implying that composition with p is equal to composing with q, implying $p = q$. $\square$

2 Positive Results

Definition 3. I say a type A is *reduced-presentable as a groupoid* if there is

$$g : \|A\|_0 \to \mathcal{U}$$

and

$$d : \forall (a : A).\, h(|a|) \simeq (a =_A a).$$

I consider this a reasonable alternative to Definition 1:

1. the higher structure of a type is fully specified by its higher loop spaces
2. consequently, in Definition 1, "too much" information is given; the HIT-construction can be done with the reduced data given in Def. 3 (of course, we will not get the general isomorphism, as both definitions would have the same problem otherwise)
3. the double-indexing in Definition 1 induces some weird choice-property that we probably would not want.

Remark 1. Assume A is reduced-presentable with (g, d). In general, it will not be presentable. In particular, it does not need to be a set. One approach to construct a presentation (h, c) would be to set

$$h(x, y) :\equiv \|x = y\|_{-1} \times g(x).$$

Intuitively, this is correct: $a = b$ is isomorphic to $a = a$ if $a = b$, and empty if $\neg a = b$. However, we will not be able to construct the general c. What we could do it offering

$$c' : \forall (a_1\, a_2 : A). \left\| h(|a_1|, |a_2|) \simeq (a_1 =_A a_2) \right\|_{-1}.$$

(if I am not mistaken, this requires LEM_{-1} in addition). The truncation avoids the "weird choice property" mentioned above.

Let's discuss some "positive" result.

Definition 4. Say that a type A has *braided* loop spaces if, for any $a : A$, the loop space $a = a$ is commutative ($\forall p, q. p \cdot q = q \cdot p$).

Caveat: Saying that $a = a$ is abelian in this case is maybe not appropriate as it does not take the higher structure into account.

The arguable most obvious examples of types with braided loop spaces are sets (for which it is trivial), and loop spaces themselves (by the Eckmann-Hilton argument).

Theorem 5. *Every type with braided loop spaces is reduced-representable.*

For an elegant proof of this, we can use the following special case of Capriotti's and my more general work:

Lemma 6 (out forthcoming work [1])**.** *Given a function $f : A \to B$ such that ap_f is constant (in the most naive sense). Under the assumption of HITs, f factors through $\|A\|_0$.* □

Proof of Theorem 5. We want to show that $f \equiv \lambda a : A.a = a$ factors through $\|A\|_0$. Thus, we need to show that ap_f is constant. Observe that ap_f is "conjugation": given $p : a = b$, the term ap_f induces an equivalence between $a = a$ and $b = b$ which is given by $\lambda q. p^{-1} \cdot q \cdot p$, and the property of having braided loop spaces is exactly what we need to say that this function is constant. □

References

[1] Paolo Capriotti and Nicolai Kraus. "Eliminating Higher Truncations via Constancy". in preparation. 2014.