# A family of types that fail to be provably collapsible

1st April 2013

This note is an informal proof *in* MLTT (intensional Martin-Löf type theory), with a conclusion *about* MLTT, of material developed formally in Agda notation in the references given below. For more discussion of the notions, see *Generalizations of Hedberg's Theorem*, to appear in TLCA'2013, by Kraus, Escardó, Coquand, and Altenkirch, but we try to be self contained.

## 1 Introduction

A type is called collapsible if it has a constant endomap, where a map is called constant if any two of its values are equal. Hence any empty or inhabited type is collapsible, and all types are collapsible under excluded middle.

We give an explicit counter-example to collapsibility. Given a type $X$ and $a_0, a_1 : X$, the type

$$(a_0 = x) + (a_1 = x)$$

can't be collapsible for all $x : X$, unless $a_0 = a_1$ is decidable. Because there are types that don't provably have decidable equality, the above family can't be provably collapsible in general.

We formulate this positively, and in type theory, as saying that if the above type is collapsible for all $x : X$ then $a_0 = a_1$ is decidable. This can be seen as a technique for proving the decidable equality of two given elements, but at present it is hard to imagine a situation where this technique would be profitably applicable.

It is an open problem to produce a single type (in a context) that can't be proved to be collapsible. This is related to the main open problem discussed in the TLCA paper.

## 2 Our version of MLTT and notation

We work in basic intensional MLTT. We don't need a universe of types. We need at least $\prod$ and $\sum$ types, identity types, and a type 2 of binary numbers 0 and 1. Given a type $X$ and elements $x, y : X$, we write $x = y$ for the identity type $\mathrm{Id}\, X x y$, leaving $X$ implicit.

We don't use HoTT (homotopy type theory) axioms, but we do use some HoTT notions and techniques. In particular, we work with the notion of hproposition (type such that any two of its elements are equal), and hpropositional truncation. It is an open problem whether such truncations are definable in MLTT, but they are under the assumption of collapsibility. Recall that an hset is a type such that $x = y$ is an hproposition for any two of its elements $x$ and $y$.

## 3 Kraus Lemma and hpropositional truncation

Kraus Lemma says that if $\kappa : X \to X$ is constant, then the type

$$\operatorname{fix}\kappa := \sum_{x:X} x = \kappa x$$

of fixed points of $\kappa$ is an hproposition. See the TLCA paper or the Agda files for the proof. This is clear if $X$ is an hset, and this particular case is enough for our counter-example, but it is interesting that the hset assumption is not needed, and we get a sharper result without this assumption. The surprising, and difficult, aspect of Klaus Lemma is that $X$ is not assumed to be an hset.

We of course define

$$\operatorname{constant}\kappa := \prod_{x,y:X} \kappa x = \kappa y.$$

From any $p : \operatorname{constant}\kappa$, we get maps

$$\begin{array}{rcl} |-| : X & \to & \operatorname{fix}\kappa \\ x & \mapsto & (\kappa x, px(\kappa x)) \\[1ex] \operatorname{fix}\kappa & \to & X, \\ (x, -) & \mapsto & x. \end{array}$$

This means that $X$ is logically equivalent to the hproposition $\operatorname{fix}\kappa$ (and $\operatorname{fix}\kappa$ is a retract of $X$). In general, if $X'$ is an hproposition logically equivalent to $X$, then $X \to X'$, and if $X \to P$ for any hproposition $P$, then $X' \to P$. That is, $X \to X'$ is the universal solution to the problem of mapping $X$ into an hproposition. Hence if $X$ is collapsed by $\kappa$ and $p$ as above, we can define its hpropositional truncation (or reflection) as

$$\|X\| := \operatorname{fix}\kappa.$$

We will write

$$\exists_{x:X}, Ax := \left\| \sum_{x:X} Ax \right\|$$

when the type $\sum_{x:X} Ax$ is collapsible.

## 4 The counter-example

Let $X$ be any type and $a_0, a_1 : X$ be arbitrary. We show that if the type

$$(a_0 = x) + (a_1 = x)$$

is collapsible for all $x : X$, then $a_0 = a_1$ is decidable. For (technical and conceptual) convenience in the proof, we regard the two elements as a single map

$$a : 2 \to X,$$

and we represent the displayed type as

$$\sum_{i:2} a_i = x.$$

The image of the map $a : 2 \to X$ is

$$E := \sum_{x:X} \exists_{i:2}, a_i = x.$$

The existential quantifier exists by the assumption that the type $\sum_{i:2}, a_i = x$ is collapsible, as discussed in the previous section. We have a map of the type 2 into the image $E$ of the function $a : 2 \to X$, defined, as usual, by

$$\begin{array}{rcl} r : 2 & \to & E \\ i & \mapsto & (a_i, |i, \mathrm{refl}\,|). \end{array}$$

Thanks to the use of hpropositional truncation,

$$r_i = r_j \iff a_i = a_j.$$

If we define $i \sim j$ to mean $a_i = a_j$, the type $E$ is the quotient of 2 by this equivalence relation (images are quotients).

The assumption of collapsibility strengthens the quotient to a retract. In fact, by the previous section, the collapsibility of the type $\sum_{i:2} a_i = x$ gives

$$(\exists_{i:2}, a_i = x) \to \sum_{i:2} a_i = x.$$

This looks like a choice function, but is simply projection in our situation. The choice has already been performed by the assumption of collapsibility. Using this, we get a map

$$s : E \to 2,$$

because there is a map $(\sum_{x:X} \sum_{i:2}, a_i = x) \to 2$ given by projection. Again crucially relying on the use of hpropositional truncation, clearly

$$r(se) = e.$$

Putting this together, we have $a_i = a_j \iff s(r_i) = s(r_j)$ and, in particular,

$$a_0 = a_1 \iff s(r_0) = s(r_1).$$

But the equality in the right-hand side is on the type 2, which is decidable. Therefore $a_0 = a_1$ is decidable.

# 5 References


1. *Generalizations of Hedberg's Theorem*, to appear in TLCA'2013, by Kraus, Escardó, Coquand, Altenkirch.

   `http://www.cs.bham.ac.uk/~mhe/papers/hedberg.pdf`

2. `http://www.cs.bham.ac.uk/~mhe/GeneralizedHedberg/html/GeneralizedHedberg.html`

   The proofs of that paper and more.

3. `http://www.cs.bham.ac.uk/~mhe/GeneralizedHedberg/non-collapsible-type/non-collapsible-type.html`

   The proof presented here.

4. `http://www.cs.bham.ac.uk/~mhe/GeneralizedHedberg/GlobalChoice/GlobalChoice.html`

   A different presentation of the same result, which emphasizes that the collapsibility of all types can be seen as a global-choice principle.

5. `http://www.cs.bham.ac.uk/~mhe/GeneralizedHedberg/ConstantChoice/ConstantChoice.html`

   This last file shows that if every type has a constant endomap, then every relation has a functional subrelation with the same domain.