Homework 3: Truncation Levels

Due 2019/03/20 (Fri) Anywhere on Earth

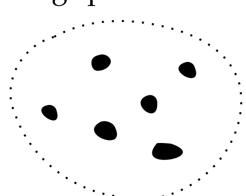
The goal of this homework is to taste a bit of *truncation levels* (homotopy levels shifted by two), one of the most important concepts in higher-dimensional type theory (and homotopy theory). We are going to prove the truncation levels of many types we have seen in Agda.

In homotopy theory, a space is at truncation level n if there is no interesting way to fold a sphere of dimension higher than n into that space. Vladimir Voevodsky found an ingenious way to express this concept in type theory as the `has-level` we saw in Agda exercises. For convenience, we call types at level n as *n-types*. The homotopy-theoretic definition looks intimidating, but cases where n is 0 or 1 can be easily visualized.

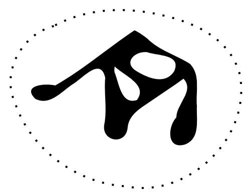
> `03/01 11:00am` "sphere of dimension higher than $n+1$" should have been "sphere of dimension higher than n".

Types at level 0 are *set-like*, which means all of its components can be contracted into points. There is only one way to fold the circle (the 1-sphere) or any space into a point—squashing the entire shape into a single spot. Folding a shape into a contractible type is not much different from folding that into a point: every folding can be identified with another; in particular, every folding can be identified with the trivial one.

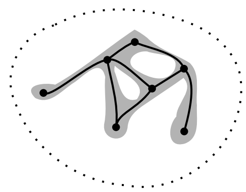
As an example, the following space is at level 0:



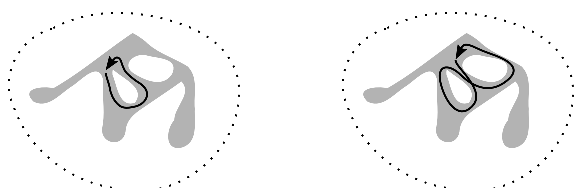
Types at the next level are *groupoid-like*. A groupoid is like a reflexive, undirected multi-graph but with potentially infinite numbers of vertices and edges. Here is a type at level 1 (1-type):



It is at level 1 because, intuitively, we can deform it into a graph:



This is not a 0-type because we have (many) non-trivial ways to fold the circle (the 1-sphere) into this space:



These two foldings are not identifiable because of the holes in the middle.

Types at the next level correspond to 2-groupoids, the 2-dimensional generalization of the (1-)groupoids. It is, however, challenging to visual them on paper. In general, one can define the type of all n-types as follows:

$$n\texttt{-types} :\equiv \sum_{A:\mathcal{U}} \texttt{has-level}_n(A)$$

where $\texttt{has-level}_n(A)$ was given in the earlier Agda exercises.

Note that the truncation level is an *upper bound* of the complexity of a type. A type at level n is always at level $n+1$. That is, the hierarchy of n-types is cumulative. There are also types which are so complicated that they are not bound by any finite level. The surface of a ball (2-sphere), for example, is not an n-type for any n. (This result might not be worked out in type theory yet.)

Negative levels -1 and -2 cannot be easily explained by the folding intuition, but they are nonetheless significant in type theory and homotopy theory. Types at level -2 are contractible to a point, which means they are equivalent to the unit type. Types at level -1 are subsingleton types (mere propositions) which can have at most one element (up to identification).

As a side note, many people refer to (mere) propositions as *properties* because we often do not care about the witness (proof) of such a property beyond its existence. However, it is a fallacy that we never care. For example, whether a function is an equivalence is a property according to this definition, but we *do* care about the inverse function embedded in the witness of some function being an equivalence.

A good way to internalize these levels is to check the levels of the types we know of. Now, load the Agda file `hw3-handout.agda` in your favorite editor and prove the following theorems. Please send the completed file to Favonia.

- The unit type itself is contractable.
- The empty type is a (mere) proposition.
- `true` is not identifiable with `false`. We will use this lemma to prove that the universe $\mathcal{U}$ is not a set.
- If A and B are n-types, then $A \times B$ is an n-type.
- If B is a family of n-types indexed by A, then $\prod_{x:A} B(x)$ is an n-type. (That is right. We do not care about the truncation level of A.)

One can also show that, if A is an n-type and B is a family of n-types, then $\sum_{x:A} B(x)$ is an n-type. Email Favonia if you use Agda to kill time.

Grading: As usual, only one letter grade (without plus or minus) will be assigned to the *entire* homework according to the criterion explained in the syllabus.