CSCI 8980 Higher-Dimensional Type Theory Lecture Notes

Dawn Michaelson, Jack Perisich

February 25, 2020

1 Universe

To internalize the $\Gamma \vdash A$ `type` judgment, we introduce the universe type $\mathcal{U}$. This can be viewed as the "type of types". We replace the original judgment with $\Gamma \vdash A : \mathcal{U}$. We can make this replacement in all our existing rules. For example, we originally had the following rule for dependent function types:

$$\frac{\Gamma \vdash A \ \texttt{type} \qquad \Gamma, x : A \vdash B \ \texttt{type}}{\Gamma \vdash \prod_{x:A} B \ \texttt{type}}$$

We can replace each judgment with a judgment that the given type has the type of the universe, giving us a new rule:

$$\frac{\Gamma \vdash A : \mathcal{U} \qquad \Gamma, x : A \vdash B : \mathcal{U}}{\Gamma \vdash \prod_{x:A} B : \mathcal{U}}$$

Doing this in all rules gives a system where we do not need the original is-type judgment.

2 Inconsistency

Because $\mathcal{U}$ is a type, we should have a rule:

$$\frac{}{\Gamma \vdash \mathcal{U} : \mathcal{U}}$$

With this rule, we can prove inconsistency, meaning a closed term of the empty type. This is the **Burali-Forti Paradox**.

1. Consider types with a transitive, well-founded order. We will write this as well-founded(A). This means that the type has a minimal element in any non-empty subset.

2. We define a type $W := \sum_{A:\mathcal{U}}$ well-founded(A). This is a type which contains all well-founded types.

3. This type W is well-founded with embedding as the relation.

   - An embedding is an order-preserving function with a strict upper bound. If we have types with ordering relations $(A, <_A)$ and $(B, <_B)$ with an embedding $f : A \to B$, f will map ordered elements from A to the same order in B, and there is some b such that $f(a) <_B b$ for any a.

   We can prove an embedding is transitive and well-founded. Then we have $(W, \text{embedding}) : W$.

4. Every well-founded type $(A, <_A)$ can be embedded into W. We can define a map $a \mapsto \sum_{a':A} a' <_A a$.

5. Because of the previous point, W can then be embedded into itself, yielding $W <_W W$. This contradicts that embedding is well-founded.

This was originally done in type theory by Girard [2]. It was later improved by Coquand [1] and Hurkens [3].

3 Repairing Universes

To remove this inconsistency, the universe $\mathcal{U}$ becomes a hierarchy of universes $\mathcal{U}_0, \mathcal{U}_1, \mathcal{U}_2, \ldots$. We have two rules for these universes:

$$\frac{}{\Gamma \vdash \mathcal{U}_i : \mathcal{U}_{i+1}} \qquad \frac{\Gamma \vdash A : \mathcal{U}_i}{\Gamma \vdash A : \mathcal{U}_{i+1}}$$

A universe is contained in the next universe above it, and the hierarchy is cumulative, so any type is contained in all the universes above it.

How does this work when a rule has multiple premises judging that types are members of some universe? Above, before introducing a universe hierarchy, we had a rule for dependent function types:

$$\frac{\Gamma \vdash A : \mathcal{U} \qquad \Gamma, x : A \vdash B : \mathcal{U}}{\Gamma \vdash \prod_{x:A} B : \mathcal{U}}$$

There are two options for writing such a rule in the hierarchy of universes. The first is to take the least upper bound of the indices of the premises ($i \sqcup j$) as the index of the conclusion:

$$\frac{\Gamma \vdash A : \mathcal{U}_i \qquad \Gamma, x : A \vdash B : \mathcal{U}_j}{\Gamma \vdash \prod_{x:A} B : \mathcal{U}_{i \sqcup j}}$$

The second is to have the same index for both premises and the conclusion:

$$\frac{\Gamma \vdash A : \mathcal{U}_i \qquad \Gamma, x : A \vdash B : \mathcal{U}_i}{\Gamma \vdash \prod_{x:A} B : \mathcal{U}_i}$$

Because we have the rule for raising a type from one universe to the next, we can raise one premise to match the other, then use this rule, so neither rule is more powerful, and it is simply a stylistic choice.

With this hierarchy and this principle for creating judgments of types being in universes, we are unable to embed W in itself, only in a version of W for a higher universe. This is because the definition of the type W requires the following rule for placing it in a universe, once we have the universe hierarchy:

$$\frac{\Gamma \vdash \mathcal{U}_i : \mathcal{U}_j \qquad \Gamma, A : \mathcal{U}_i \vdash \text{well-founded}(A) : \mathcal{U}_j}{\Gamma \vdash \sum_{A:\mathcal{U}_i} \text{well-founded}(A) : \mathcal{U}_j}$$

Because $\mathcal{U}_i$, the type of A, is an element of $\mathcal{U}_j$, $i < j$, and this dependent pair type cannot be an element of itself. Because of this, we cannot recreate the Burali-Forti Paradox with these rules.

In practice, proofs which do not distinguish between universes in the hierarchy, acting as though the judgment $\Gamma \vdash \mathcal{U} : \mathcal{U}$ holds, can usually have universe levels inserted consistently. For this reason, indices are often not shown when they do not affect the proof.

4 Univalence Principle

The univalence principle is roughly that equivalent things should be identified. A good approximation of this is:

$$(A \simeq B) \simeq \mathtt{Id}_{\mathcal{U}}(A; B)$$

This requires that we define equivalences $A \simeq B$. We define this as

$$(A \simeq B) := \sum_{f:A\to B} \text{is-equiv}(f)$$

This, in turn, requires us to give a good definition of is-equiv().

4.1 Function Equivalence

A quasi-inverse of a function $f : A \to B$ (qinv(f)) includes

- $g : B \to A$
- $\epsilon : \prod_{y:B} \mathtt{Id}_B(f(g(y)); y)$
- $\eta : \prod_{x:A} \mathtt{Id}_A(g(f(x)); x)$

This may be formulated as a rather large dependent pair type.

For a good definition of is-equiv(f), we want two things:

1. The definition should be logically equivalent to qinv(f) (either one can be derived given the other)
2. is-prop(is-equiv(f))
   This requires that is-equiv(f) has at most one element. The definition of qinv(f) fails this requirement.

We give two possible types to fulfill the requirements on is-equiv(f):

1. Half Adjoint Equivalence (is-hae(f))

   This includes g, ϵ, and η as in the quasi-inverse, but it also includes a fourth element τ:

   $$\tau : \prod_{x:A} \mathtt{Id}(\mathrm{ap}_f(\eta(x)); \epsilon(f(x)))$$

   This τ may be thought of as ensuring that η and ϵ are coherent in some way.

   It is clear how to derive qinv(f) given is-hae(f), since we already have the g, ϵ, and η. The other direction of the equivalence is much more difficult to prove.

Rather than τ, we could have defined τ':

$$\tau' : \sum_{y:B} \mathtt{Id}(\eta(g(y)); \mathrm{ap}_g(\epsilon(y)))$$

This definition is dual to the definition of τ. It is the other "half" in the name of the type. If the definition required both τ and τ', it would not satisfy the requirement of is-prop(is-equiv(f)), and we would require a further element to show that τ and τ' were coherent.

2. is-equiv(f) := $\prod_{y:B}$ is-contr($\sum_{x:A} \mathtt{Id}(f(x); y)$)

   The inner dependent pair type is defining the pre-image of y (in homotopy theory, this is the "fiber over y" or that "all fibers are contractible"). There is only one element in the pre-image, which gives an inverse function.

Both of these definitions are acceptable. In Agda, we prefer half adjoint equivalence because it is easier to use.

A difficulty is that proving is-prop(is-equiv(f)) for many definitions requires function extensionality, which cannot be proved without univalence. We choose a definition of is-equiv() and use it to define univalence, then prove is-prop() for it assuming univalence.

4.2 Univalence and Function Extensionality

A precise formulation of univalence:

> Define idtoequiv : $\mathtt{Id}_{\mathcal{U}}(A; B) \to A \simeq B$
> Axiom: is-equiv(idtoequiv)
> idtoequiv := $\lambda p.\mathtt{J}(A.$a proof of is-equiv(id_A); p)

We define function extensionality as follows, using it to prove is-prop(is-equiv()).

> (Strong) Function Extensionality
> Define happly : $\mathtt{Id}(f; g) \to \prod_{x:A} \mathtt{Id}(f(x); g(x))$
> Axiom: is-equiv(happly)

Some points to note:

1. By including these two axioms, we are breaking harmony. This is fixed by cubical type theory.

2. Univalence can be viewed as an extensionality principle for universes.
3. Univalence implies function extensionality, so we don't need the axiom for happly.
4. Univalence implies ¬axiom k and ¬LEM. If we had the law of the excluded middle, that would imply that every type has decidable equality, which would in turn imply that every type is a set. This is a contradiction, since $\mathcal{U}$ is not a set.

References

[1] Thierry Coquand. "An Analysis of Girard's Paradox". In: *In Symposium on Logic in Computer Science*. IEEE Computer Society Press, 1986, pp. 227–236.

[2] J. Girard. "Interprétation fonctionelle et élimination des coupures de l'arithmétique d'ordre supérieur". PhD thesis. Université Paris VII, 1972.

[3] Antonius J. C. Hurkens. "A simplification of Girard's paradox". In: *Typed Lambda Calculi and Applications*. Ed. by Mariangiola Dezani-Ciancaglini and Gordon Plotkin. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 266–278. ISBN: 978-3-540-49178-1.