An Inductive Universe for Setoids

Loïc Pujet

Stockholm University, Sweden
loic@pujet.fr

Setoid models of type theory were designed by Hofmann [5] in order to add function extensionality, proposition extensionality, and quotient types to intensional type theory. Unfortunately, Hofmann's first setoid model does not support true type dependency, and his second setoid model does not support all the computation rules of type theory. Altenkirch [1] improved the definition of setoids by putting the setoid equalities in a sort of *strict* propositions (`SProp`), which allowed him to support dependent types. A few years later down the line, Altenkirch, Boulier, Kaposi, Sattler and Sestini constructed a universe of setoids for Altenkirch's model [2]. They give three different definitions of their universe: one which needs *induction-recursion*, one which needs *induction-induction*, and finally one which only requires an identity type with a strengthened J rule.

In this abstract, we propose a new construction for the setoid universe which only needs indexed inductive families. Furthermore, we note that our construction still works (more or less) when the equality of the setoids is taken to be in `Prop` or `Type` instead of `SProp`, resulting in models that can be equipped with *choice principles*. Our construction is fully formalised in Rocq, and is available at https://github.com/loic-p/setoid-universe.

1 The Universe of Setoids

The most natural construction for the setoid universe is *via* induction-recursion: one defines a type of codes $\mathtt{U}_0$ along with three recursive functions $\mathtt{eqU}_0$, $\mathtt{El}_0$ and $\mathtt{eq}_0$ that respectively represent the setoid equality between the codes, the universal family of small setoids and its (heterogeneous) equality:

$$\mathtt{U}_0 : \mathtt{Type}_1 \qquad\qquad \mathtt{El}_0 : \mathtt{U}_0 \to \mathtt{Type}_0$$
$$\mathtt{eqU}_0 : \mathtt{U}_0 \to \mathtt{U}_0 \to \mathtt{Sort}_1 \qquad\qquad \mathtt{eq}_0 : \forall\ (\mathtt{A}\ \mathtt{B} : \mathtt{U}_0),\ \mathtt{El}_0\ \mathtt{A} \to \mathtt{El}_0\ \mathtt{B} \to \mathtt{Sort}_0$$

Note that we use an indeterminate sort $\mathtt{Sort}_i$ for the equality relations of our setoids, so that we may later instantiate it with either `SProp`, `Prop` or $\mathtt{Type}_i$. Anyway, we do not wish to use induction-recursion, so we must find a way to eliminate it from the construction. The canonical method is Hancock *et al.*'s small induction-recursion [4], but it does not apply in this case: not only do the recursive functions $\mathtt{eqU}_0$ and $\mathtt{eq}_0$ have two arguments of type $\mathtt{U}_0$ instead of one, but the return type of $\mathtt{eqU}_0$ is not even small (at least in the case where $\mathtt{Sort}_i = \mathtt{Type}_i$).

Instead, we start by defining an overapproximation $\mathtt{preU}_0$ for our setoid universe, with a constructor for dependent products that is parametrised by arbitrary equality relations on `A` and `P` and does not enforce `P` to be a setoid morphism (Fig. 1). This way, the definition of $\mathtt{preU}_0$ is not mutual with the definition of $\mathtt{eq}_0$ and $\mathtt{eqU}_0$ anymore, and it fits in the framework of small induction-recursion. Then, as a second step, we define the equality relations $\mathtt{eq}_0$ and $\mathtt{eqU}_0$ on the overapproximated universe, using the same definitions as in the usual inductive-recursive version. Next, we define an inductive predicate $\mathtt{extU}_0$ that carves out the codes from $\mathtt{preU}_0$ which have a counterpart in the inductive-recursive definition. More specifically, it ensures that that the dependent codomains that appear in $\mathtt{pre}_\Pi$ and $\mathtt{pre}_\Sigma$ are proper setoid morphisms from the domains into the universe, and that the codes for dependent products have

```
Inductive preU₀ : Type₁ :=
| pre_ℕ : preU₀
| pre_Σ : ∀ (A : preU₀) (P : El₀ A → preU₀), preU₀
| pre_Π : ∀ (A : preU₀) (Aeq : El₀ A → El₀ A → Sort₀)
          (P : El₀ A → preU₀) (Peq : ∀ a₀ a1, El₀ (P a₀) → El₀ (P a₁) → Sort₀), preU₀.

Fixpoint El₀ (A : preU₀) : Type₀ :=
  match A with
  | pre_ℕ ⇒ ℕ
  | pre_Σ A P ⇒ Σ (a : El₀ A), El₀ (P a)
  | pre_Π A Aeq P Peq ⇒ Σ (f : ∀ (a : El₀ A), El₀ (P a)), (∀ a a', Aeq a a' → Peq (f a) (f a'))
  end.
```

Figure 1: Small inductive-recursive definition of an overapproximated universe.

been parametrised with the equality relations defined by $\mathtt{eq}_0$. Finally, we can put everything together: the carrier type of our universal setoid is defined as $\mathtt{U}_0 := \Sigma\ (\mathtt{A} : \mathtt{preU}_0).\ \mathtt{extU}_0\ \mathtt{A}$, its setoid equality is given by $\mathtt{eqU}_0$ on the first component, the universal dependent family on the universe is provided by $\mathtt{El}_0$, and the heterogeneous equality on that family is given by $\mathtt{eq}_0$. This roundabout encoding is actually faithful to the original inductive-recursive definition, as we can derive the same induction principle with its computation rules, and the three functions $\mathtt{El}_0$, $\mathtt{eqU}_0$ and $\mathtt{eq}_0$ compute on type formers.

In order to complete the definition of our universal setoid, we can show by induction that the equality relation $\mathtt{eqU}_0$ is an equivalence relation on $\mathtt{U}_0$, and that the relation $\mathtt{eq}_0$ is a heterogeneous equality equipped with a coercion operator:

$$\begin{aligned}
\mathtt{cast} &: \forall\ (\mathtt{A}\ \mathtt{B} : \mathtt{U}_0)\ (\mathtt{e} : \mathtt{eqU}_0\ \mathtt{A}\ \mathtt{B})\ (\mathtt{a} : \mathtt{El}_0\ \mathtt{A}),\ \mathtt{El}_0\ \mathtt{B} \\
\mathtt{casteq} &: \forall\ (\mathtt{A}\ \mathtt{B} : \mathtt{U}_0)\ (\mathtt{e} : \mathtt{eqU}_0\ \mathtt{A}\ \mathtt{B})\ (\mathtt{a} : \mathtt{El}_0\ \mathtt{A}),\ \mathtt{eq}_0\ \mathtt{A}\ \mathtt{B}\ \mathtt{a}\ (\mathtt{cast}\ \mathtt{A}\ \mathtt{B}\ \mathtt{e}\ \mathtt{a})
\end{aligned}$$

The accompanying Rocq development also includes W types, a subuniverse of propositions, an equality type, two universe levels, quotient types, and an accessibility predicate with its large elimination principle.

2 Choice principles

SProp setoids Instantiating the construction with `SProp` results in a universe that fits nicely in Altenkirch's setoid model, and does not need anything fancy from our metatheory. This provides a shallow embedding of MLTT + extensionality + quotients into MLTT + SProp, which preserves *all* the computation rules of MLTT. Note that even though the computation rule of the J eliminator for $\mathtt{eq}_0$ is only propositional, one can nevertheless define an inductive equality in $\mathtt{U}_0$ which is equivalent to $\mathtt{eq}_0$ and for which J does compute on reflexivity [8].

Type setoids The situation is even more interesting if we try to instantiate the construction with `Type`. In that case, we can additionally interpret the following principle in our model, which says that any relation that is propositionally a functional relation determines a function (this principle is sometimes called *unique choice*):

$$\begin{aligned}
&\forall\ (\mathtt{a} : \mathtt{A}),\ \|\ \Sigma\ (\mathtt{b} : \mathtt{B}),\ (\mathtt{R}\ \mathtt{a}\ \mathtt{b}) \times (\forall\ \mathtt{c},\ \mathtt{R}\ \mathtt{a}\ \mathtt{c} \to \mathtt{c} = \mathtt{b})\ \| \\
&\quad \to\ \Sigma\ (\mathtt{f} : \mathtt{A} \to \mathtt{B}),\ \forall\ (\mathtt{a} : \mathtt{A}),\ \mathtt{R}\ \mathtt{a}\ (\mathtt{f}\ \mathtt{a}).
\end{aligned}$$

And there's more: since the equality relation on the setoid of natural numbers coincides with the meta-theoretic equality on its underlying set, a function from $\mathbb{N}$ to any other setoid is automatically a setoid morphism. As a consequence, one can interpret countable choice and even dependent choice in this new model.

$$\begin{array}{lll} \mathsf{AC}_{\mathbb{N}} & : & (\forall\ (\mathtt{n} : \mathbb{N}),\ \|\ \mathtt{P\ n}\ \|) \ \rightarrow\ \|\ \forall\ (\mathtt{n} : \mathbb{N}),\ \mathtt{P\ n}\ \| \\ \mathsf{DC} & : & (\forall\ (\mathtt{a} : \mathtt{X}),\ \|\ \Sigma\ (\mathtt{b} : \mathtt{X}),\ \mathtt{R\ a\ b}\ \|) \ \rightarrow\ \|\ \Sigma\ (\mathtt{s} : \mathbb{N} \rightarrow \mathtt{X}),\ \forall\ \mathtt{n},\ \mathtt{R\ n}\ (\mathtt{n}{+}\mathtt{1})\ \| \end{array}$$

Unfortunately, there is a price to pay: if we try to do a setoid model without definitional proof irrelevance, the computation rules for substitution under binders do not hold definitionally. In particular, this means that our construction can no longer be viewed as a shallow embedding from an extension of MLTT to MLTT, since some definitional equalities are only interpreted as setoid equalities. This phenomenon has already been pointed out in a note of Coquand [3], in which he presents proof-relevant setoids as a model for an early version of MLTT which had weaker computation rules [6, 7].

Prop setoids If we instantiate our construction with `Prop` instead, we can have a sort of impredicative propositions without destroying the computational content of propositions. In particular, it allows us to have large elimination for the `Prop`-valued accessibility predicate. As a consequence, we can derive a version of unique choice which is restricted to decidable relations $\mathtt{R} : \mathtt{A} \rightarrow \mathbb{N} \rightarrow \mathtt{Prop}$. The resulting theory is extensional but keeps the proof-theoretic strength of CIC, which is much higher than the strength of MLTT+SProp. However, neither the full principle of unique choice nor countable choice are provable in this model, and the computation rules for binders are not definitional either.

References

[1] Thorsten Altenkirch. Extensional equality in intensional type theory. In *Proceedings. 14th Symposium on Logic in Computer Science (Cat. No. PR00158)*, pages 412–420, 1999.

[2] Thorsten Altenkirch, Simon Boulier, Ambrus Kaposi, Christian Sattler, and Filippo Sestini. Constructing a universe for the setoid model. In Stefan Kiefer and Christine Tasson, editors, *Foundations of Software Science and Computation Structures*, pages 1–21, Cham, 2021. Springer International Publishing.

[3] Thierry Coquand. About the setoid model. https://www.cse.chalmers.se/~coquand/setoid.pdf, 2013.

[4] Peter Hancock, Conor McBride, Neil Ghani, Lorenzo Malatesta, and Thorsten Altenkirch. Small induction recursion. In Masahito Hasegawa, editor, *Typed Lambda Calculi and Applications*, pages 156–172, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[5] Martin Hofmann. A simple model for quotient types. In Mariangiola Dezani-Ciancaglini and Gordon Plotkin, editors, *Typed Lambda Calculi and Applications*, pages 216–234, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[6] Per Martin-Löf. An intuitionistic theory of types: Predicative part. In H.E. Rose and J.C. Shepherdson, editors, *Logic Colloquium '73*, volume 80 of *Studies in Logic and the Foundations of Mathematics*, pages 73 – 118. Elsevier, 1973.

[7] Per Martin-Löf. About models for intuitionistic type theories and the notion of definitional equality. In Stig Kanger, editor, *Proceedings of the Third Scandinavian Logic Symposium*, volume 82 of *Studies in Logic and the Foundations of Mathematics*, pages 81–109. Elsevier, 1975.

[8] Loïc Pujet and Nicolas Tabareau. Observational Equality: Now For Good. *Proceedings of the ACM on Programming Languages*, 6(POPL):1–29, January 2022.