Extensional Finite Sets and Multisets in Type Theory

Clemens Kupke Fredrik Nordvall Forsberg **Sean Watters**

University of Strathclyde

13/06/2024

Motivation

We want a data type for collections of unordered data (ie, finite sets and multisets), which:

- Has decidable equality iff the underlying type does.
- Satisfies the expected equational theories.
- Works in “standard” MLTT.

Notions of Subsets and Multisets in Type Theory

In short:

- Given some S : Set, subsets of S are unary predicates $S \to$ Prop.
- *Decidable* subsets are functions $S \to 2$.
- Multisets over S are functions $S \to \mathbb{N}$.

Notions of Subsets and Multisets in Type Theory

In short:

- Given some S : Set, subsets of S are unary predicates $S \to \text{Prop}$.
- *Decidable* subsets are functions $S \to 2$.
- Multisets over S are functions $S \to \mathbb{N}$.

Desirable properties:

- Extenensionality:

Notions of Subsets and Multisets in Type Theory

In short:

- Given some S : Set, subsets of S are unary predicates $S \to$ Prop.
- *Decidable* subsets are functions $S \to 2$.
- Multisets over S are functions $S \to \mathbb{N}$.

Desirable properties:

- Extenensionality: $(X = Y) \iff (\forall x.\ x \in X \iff x \in Y)$

Notions of Subsets and Multisets in Type Theory

In short:

- Given some S : Set, subsets of S are unary predicates $S \to$ Prop.
- *Decidable* subsets are functions $S \to 2$.
- Multisets over S are functions $S \to \mathbb{N}$.

Desirable properties:

- Extenensionality: $(X = Y) \iff (\forall x.\ x \in X \iff x \in Y)$
- Decidable Equality

Notions of Subsets and Multisets in Type Theory

In short:

- Given some S : Set, subsets of S are unary predicates $S \to$ Prop.
- *Decidable* subsets are functions $S \to 2$.
- Multisets over S are functions $S \to \mathbb{N}$.

Desirable properties:

- Extenensionality: $(X = Y) \iff (\forall x.\ x \in X \iff x \in Y)$
- Decidable Equality, which we would expect to follow from finiteness.

Finiteness, Decidable Equality, Extensionality

What approaches are available for finite subsets?

- $S \to 2$?
  (No decidable equality.)
- An enumeration list?
  (No extensionality.)
- A higher-inductive type? (Choudhury & Fiore, 2023; Joram & Veltri, 2023)
  (Works, but restricts us to HoTT.)
- A sorted list?
  (Our approach; need to treat the ordering data with care. See also: Appel & Leroy, 2023; Krebbers, 2023; the Rocq libraries `fset`, `extructures`, `finmap`, `ssrmisc`.)

The Equational Theory of Finite Sets

We expect notions of union and empty set, satisfying:

- $X \cup \varnothing = \varnothing \cup X = X$ (unit).
- $X \cup X = X$ (idempotency).
- $X \cup Y = Y \cup X$ (commutativity).
- $X \cup (Y \cup Z) = (X \cup Y) \cup Z$ (associativity).

Theorem (Folklore?)

In the context of set theory, the finite powerset $\mathscr{P}_f(X)$, is the free idempotent commutative monoid over the set X.

We show that the free idem. comm. monoid is realised in type theory by sorted lists.

Fresh Lists

We study sorted lists as an instance of the following generalisation:

$$
\begin{array}{l}
\text{mutual} \\
\quad \text{data } \mathsf{FList} \ \{X : \mathsf{Set}\} \ (R : X \to X \to \mathsf{Set}) : \mathsf{Set} \text{ where} \\
\qquad \mathsf{nil} : \mathsf{FList}\ R \\
\qquad \mathsf{cons} : (x : X) \to (xs : \mathsf{FList}\ R) \to x \# xs \to \mathsf{FList}\ R
\end{array}
$$

$$
\begin{array}{l}
\quad \_\#\_ : \{X : \mathsf{Set}\} \ \{R : X \to X \to \mathsf{Set}\} \\
\qquad\quad \to X \to \mathsf{FList}\ R \to \mathsf{Set} \\
\quad x \# \mathsf{nil} = \top \\
\quad \_\#\_ \ \{R = R\} \ x \ (\mathsf{cons}\ y\ ys\ p) = (R\ x\ y) \times (x \# ys)
\end{array}
$$

Originally due to Catarina Coquand. Generalisation to an arbitrary R due to the Agda standard library.

Sorted Lists

Sorted lists (without duplicates) arise as fresh lists over an irreflexive total order $<$:

- The ordering ensures that any two lists with the same elements are equal.
- Irreflexivity forces any given element to appear exactly once in any given list.

Monoid Structure

- Unit: The empty list.
- Multiplication: Merge sort (defined by recursion on the lists, using totality of $<$).

Extensionality Principle

Proving that the laws of $\varnothing$ and $\cup$ hold for sorted lists by induction is messy. Instead:

Theorem: The Extensionality Principle for Sorted Lists

For all $xs, ys : \mathsf{FList}(X, <)$:

$$xs = ys \text{ iff } (a \in xs) \iff (a \in ys) \text{ for all } a : X.$$

With this sledgehammer, the proofs of the equations for $\cup$ become much easier.

Theorem

$(\mathsf{FList}(X, <),\ \cup,\ \mathsf{nil})$ is an idempotent commutative monoid.

Freeness

Freeness is formulated as a universal property; sorted lists form a functor which is left adjoint to a forgetful functor. But what are the categories?

Freeness

Freeness is formulated as a universal property; sorted lists form a functor which is left adjoint to a forgetful functor. But what are the categories?

The Category STO

- Objects: Sets, equipped with strict total orders.
- Morphisms: *Not necessarily monotone* functions on the underlying sets.

The Category OICMon

- Objects: Idempotent commutative monoids, with strict total orders.
- Morphisms: *Not necessarily monotone* monoid morphisms.

Freeness

Freeness is formulated as a universal property; sorted lists form a functor which is left adjoint to a forgetful functor. But what are the categories?

The Category STO

- Objects: Sets, equipped with strict total orders.
- Morphisms: *Not necessarily monotone* functions on the underlying sets.

The Category OICMon

- Objects: Idempotent commutative monoids, with strict total orders.
- Morphisms: *Not necessarily monotone* monoid morphisms.

Theorem: The Universal Property of Ordered Idem. Comm. Monoids

SList : STO → OICMon forms a functor which is left adjoint to the forgetful functor $\mathcal{U}$: OICMon → STO defined by $\mathcal{U}(X, <, \cdot, \epsilon) := (X, <)$.

Multisets

Fresh lists over a decidable reflexive total order $\leq$ realise finite multisets.

$\in$ is prop-valued for finite sets, but set-valued for finite multisets. So we need a different extensionality principle:

Theorem: Extensionality Principle for $\mathsf{FList}(X, \leq)$

There is a "multiplicity function", $\mathsf{count} : \mathsf{FList}(X, \leq) \to X \to \mathbb{N}$, such that:

For all $a : X$, and $xs, ys : \mathsf{FList}(X, \leq)$,

$$\mathsf{count}\ xs\ a = \mathsf{count}\ ys\ a \quad \Leftrightarrow \quad (a \in xs) \cong (a \in ys) \quad \Leftrightarrow \quad xs = ys.$$

Multisets

Fresh lists over a decidable reflexive total order $\leq$ realise finite multisets.

$\in$ is prop-valued for finite sets, but set-valued for finite multisets. So we need a different extensionality principle:

Theorem: Extensionality Principle for $\mathsf{FList}(X, \leq)$

There is a "multiplicity function", $\mathsf{count} : \mathsf{FList}(X, \leq) \to X \to \mathbb{N}$, such that:

For all $a : X$, and $xs, ys : \mathsf{FList}(X, \leq)$,

$$\mathsf{count}\ xs\ a = \mathsf{count}\ ys\ a \quad \Leftrightarrow \quad (a \in xs) \cong (a \in ys) \quad \Leftrightarrow \quad xs = ys.$$

Analagous to before:

Theorem: Universal Property of Ordered Commutative Monoids

$\mathsf{SListD} : \mathsf{DTO} \to \mathsf{OCMon}$ forms a functor which is left adjoint to the forgetful functor $\mathcal{U} : \mathsf{OCMon} \to \mathsf{DTO}$.

More Free Algebraic Structures

Different notions of "freshness" yield different free algebraic structures:

Freshness Relation	Free Algebraic Structure	Data Structure
$\leq$, a total order	Ordered Commutative Monoid	Sorted lists
$<$, a strict total order	Ordered Idempotent Comm. Monoid	Sorted lists w/o duplicates
$\lambda x.\lambda y.\bot$	Pointed Set	Maybe
$\lambda x.\lambda y.\top$	Monoid	List
$\neq$	Left-Regular Band Monoid	Lists without duplicates
$=$	Reflexive Partial Monoid	$1 + (A \times \mathbb{N}^{>0})$

Summary

- We saw the data type of (generalised) fresh lists.
- We saw how they realise finite sets and multisets, and proved the relevant universal properties.
- We glimpsed the zoo of other free algebraic structures that can be represented this way.

Further reading:

- Kupke, C., Nordvall Forsberg, F., Watters, S.: *A fresh look at commutativity: free algebraic structures via fresh lists*. In: APLAS '23.

  `https://doi.org/10.1007/978-981-99-8311-7_7`
- Full Agda formalisation: `https://seanwatters.uk/agda/fresh-lists`

Bonus: Why No Monotonicity?

A few reasons:

- It breaks the adjunction.
- We get a (subjectively) more natural notion of functoriality without it.
- It's an implementation detail.
- Without it, we get a nice result relating our constructions back to classical finite (multi)sets:

If only there was a "free strict total order on a set", then we could ignore the ordering data and obtain the genuine $\mathscr{P}_f$. But such a thing is a weak form of AC called the Ordering Principle, which implies LEM. However:

Theorem

Assuming OP, Set $\cong$ STO, OICMon $\cong$ ICMon, etc.

References

- Coquand, C.: *A formalised proof of the soundness and completeness of a simply typed lambda-calculus with explicit substitutions*. `https://doi.org/10.1023/A:1019964114625`
- Choudhury, V., Fiore, M.: *Free commutative monoids in Homotopy Type Theory*. `https://doi.org/10.46298/entics.10492`
- Joram, P., Veltri, N.: *Constructive final semantics of finite bags*. `https://doi.org/10.4230/LIPIcs.ITP.2023.20`
- Appel, A.W., Leroy, X.: *Efficient extensional binary tries*. `https://doi.org/10.1007/s10817-022-09655-x`