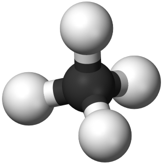
The agda-unimath library

Egbert Rijke

University of Ljubljana

2nd International HoTT Conference, May 24, 2023

Joint work with
Fredrik Bakke
Elisabeth Bonnevier
Jonathan Prieto-Cubides



Basic information on the agda-unimath library

Purpose of the agda-unimath library

- The agda-unimath library aims to be a general purpose library of formalized mathematics from a univalent point of view.
- The agda-unimath library aims to be an informative resource for mathematicians.
- The agda-unimath library contains currently only constructive univalent mathematics, but we are also open to contributions of classical univalent mathematics.

Where to find us

- Repository: `https://github.com/UniMath/agda-unimath`
- Website: `https://unimath.github.io/agda-unimath/`
- Discord: Univalent Agda (Community with over 350 members, discussing four univalent agda libraries)

Contributions in any area of mathematics are welcome.

History of the agda-unimath library

The agda-unimath library was founded on November 19th 2021 by

- Egbert Rijke
- Elisabeth Bonnevier
- Jonathan Prieto-Cubides

during the *Univalent foundations for daily applications* meeting organized by Marc Bezem. Fredrik Bakke has since then joined the team of maintainers.

The initial goal was to formalize the *Symmetry Book* by Bezem, Buchholtz, Cagne, Dundas, and Grayson, using the existing formalization of the *Introduction to Homotopy Type Theory* by Egbert Rijke. We soon realized that our library organization and topics made better sense as a general purpose library of formalized univalent mathematics.

Within a year, the agda-unimath library became the biggest library of formalized mathematics in Agda, univalent or otherwise. It has approximately 160.000 lines of code today.

Contributors

The agda-unimath library is the largest and fastest growing library in Agda, with over 200,000 lines of code over more than 1200 files, compiled into a markdown book of 3745 pages.

28 people have contributed so far (Thank you all!!)

- Egbert Rijke (393,554 ++, 228,771 --)
- Fredrik Bakke (56,093 ++, 37,534 --)
- Éléonore Mangel (25,885 ++, 12,287 --)
- Elisabeth Bonnevier (10,220 ++, 7,258 --)
- Jonathan Prieto-Cubides (108,649 ++, 102,706 --)
- Raymond Baker (2,250 ++, 1,005 --)
- Elif Uskuplu (3,409 ++, 765 --)
- Bryan Lu (4,479 ++, 2,494 --)
- Fernando Chu (6,856 ++, 912 --)
- Victor Blanchi (23,172 ++, 5,138 --)
- ...

Rough comparison with other libraries

Library	*LOC*[1]	*Files*[2]	*Contributors*[3]
lean mathlib	1,092,000	3491	306
Unimath	482,000	1091	64
agda-unimath	**223,000**	**1275**	**28**
TypeTopology	140,000	501	11
cubical	132,000	975	80
agda-stdlib	126,000	1273	>100
Coq-HoTT	117,000	621	58
1lab	81,000	540	27
yoneda	5,545	18	3

1. The number of lines of code, including empty lines.
2. The number of formalization files, not including supporting files.
3. The number of people who made commits into the library, not including other people who may be considered contributors.

Mathematical foundation of agda-unimath

The agda-unimath library is written in plain Agda, which implements a variant of Martin-Löf's dependent type theory, using the options **without-K** and **exact-split**.

Postulates in agda-unimath

- Function extensionality
- Univalence
- Truncations
- Type theoretic replacement
- The interval
- The circle
- Homotopy pushouts

Even though we have these postulates, we try to not use a postulate if it is not necessary. In particular, we try to not use univalence or replacement if we can avoid it.

Contents of the agda-unimath library

Mathematical subjects in agda-unimath

- Category theory (40)
- Commutative algebra (38)
- Elementary number theory (110)
- Finite group theory (20)
- Foundation (291)
- Graph theory (46)
- Group theory (134)
- Higher group theory (13)
- Linear algebra (18)
- Lists (18)
- OEIS (1)
- Order theory (80)
- Organic chemistry (8)
- Orth. factorization systems (21)
- Polytopes (1)
- Real numbers (1)
- Ring theory (43)
- Set theory (7)
- Species (41)
- Structured types (43)
- Synthetic homotopy theory (38)
- Trees (47)
- Type theories (27)
- Univalent combinatorics (88)
- Universal algebra (11)

Trees project (ongoing)

Consider a type family B over A. The W-type $\mathsf{W}(A, B)$ is inductively generated by one constructor

$$\mathsf{tree} : \prod_{(x:A)} (B(x) \to \mathsf{W}(A, B)) \to \mathsf{W}(A, B).$$

Question

Elements of W-types are often said to be trees. According to what concept of tree can we indeed view elements of W-types as trees?

Formalization

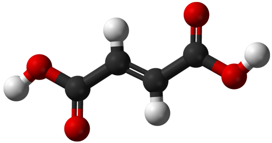
- Undirected trees (has decidable equality on nodes)
- Directed trees
- Acyclic graphs
- Enriched directed and undirected trees

Organic chemistry project

Isomerism

There are molecules that have the same underlying graph, but nevertheless we can distinguish them based on their spatial arrangement. Such pairs of molecules are called **isomers**.

The agda-unimath library has a definition of hydrocarbons in univalent mathematics in such a way that distinct isomers are correctly distinguished.



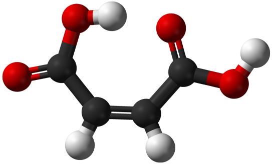


Image credit:
Ben Mills, Wikipedia (Public Domain)

Group theory

The agda-unimath library started as a formalization project for the symmetry book.

Overview

Groups appear in at least two ways in univalent mathematics:

- Groups are sets equipped with the structure of a group.
- Groups are pointed connected 1-types.

These descriptions of groups are equivalent, but give a different perspective on the theory of groups. The library develops group theory from both points of view simultaneously. Furthermore, the library contains a development of the following topics

- Semigroups
- Monoids
- Commutative monoids
- Groups
- Abelian groups

The sign homomorphism (project with Éléonore Mangel)

Since the symmetric group is the type BS_n of all n-element types, there should be a natural description of a pointed map

$$B\sigma : BS_n \to_* BS_2$$

delooping the sign homomorphism.

Cartier's method

For an n-element type X, consider the type

$$D_X := \prod_{(A:\binom{X}{2})} A$$

equipped with the equivalence relation

$$R_X(d, d') := \#\{A \mid d(A) \neq d'(A)\} \equiv 0 \mod 2.$$

Then D_X/R_X is a 2-element type and $X \mapsto D_X/R_X$ deloops the sign homomorphism.

The Zariski locale (ongoing project with Maša Zaucer)

For any commutative ring A, the Zariski locale $\mathrm{Spec}(A)$ of A is the locale of radical ideals of A.

Observations

- The Zariski locale is constructed constructively, but it is classically equivalent to the locale of opens of the Zariski topology.
- The Zariski locale contains radical ideals of all universe levels, and is therefore naturally large.

Library

In the ring theory folder we develop the theories of

- Rings
- Semirings

In the commutative algebra folder we develop the theories of

- Commutative rings
- Commutative semirings

Large posets

Definition

Consider functions $\alpha : \text{Level} \to \text{Level}$ and $\beta : \text{Level} \to \text{Level} \to \text{Level}$. An (α, β)-**indexed large poset** consists of

- For each universe level l, a type $X_l : \mathcal{U}_{\alpha(l)}$ of elements.
- For any $x : X_{l_1}$ and $y : X_{l_2}$ a proposition

$$x \leq y$$

  of universe level $\beta(l_1, l_2)$, such that
- For any $x : X_l$ we have $x \leq x$
- For any $x : X_{l_1}$, $y : X_{l_2}$, and $z : X_{l_3}$ we have

$$(y \leq z) \to (x \leq y) \to (x \leq z).$$

- For any $x, y : X_l$ we have

$$(x \leq y) \to (y \leq x) \to (x = y).$$

Large locales

Definition

A **large locale** is a large poset equipped with

- For any $x : X_{l_1}$ and $y : X_{l_2}$ a meet

  $$x \wedge y : X_{l_1 \sqcup l_2}$$

  which satisfies the universal property of the meet with respect to all universe levels.

- For any type $I : \mathcal{U}_{l_1}$ and any $x : I \to X_{l_2}$ an element

  $$\bigvee_i x_i : X_{l_1 \sqcup l_2}$$

  which satisfies the universal property of the join with respect to all universe levels.

- For any $x : X_{l_1}$ and any $y : I \to X_{l_3}$ an identification

  $$x \wedge \bigvee_i y_i = \bigvee_i x \wedge y_i.$$

Examples of large locales

Examples

- Prop is a large locale.
- Dependent products of large locales are large locales.
- $A \to$ Prop is a large locale for any type A.
- Spec(A) is a large locale for any commutative ring A.

Large locales occur naturally in the context of formalization.
Large locales are predicative and complete.

Design principles

We believe libraries of formalized mathematics could be informative resources.

- All files in the source are **literate agda markdown** files.
- The website is an **mdbook**.
- One concept per file, like on a wiki.
- Lines of code are bounded by 80 characters, like a text.
- Clear names. Code that is riddled with abbreviations is unreadable.
- Use unicode, but do so sparingly.
- **Separation of library organization from how things are formalized.**
- Avoid silly games like "use the least number of characters possible in a proof" or "minimize the number of imports". Maintain conceptual clarity, because formalized mathematics is already difficult enough.

We don't claim that these design principles are better than others. They are merely design principles that work for us, that we find practical, and that align with the goal of the agda-unimath library to be an informative resource of formalized univalent mathematics.

Thank you!

Happy formalizing!