The Blueprint For Formalizing Geometric Algebra in Lean

Eric Wieser, Utensil Song

October 18, 2023

Introduction

The goal of this document is to provide a detailed account of the formalization of Geometric Algebra (GA) a.k.a. Clifford Algebra [Hestenes and Sobczyk(1984)] in the Lean 4 theorem prover and programming language [Moura and Ullrich(2021), de Moura et al.(2015), Ullrich(2023)] and using its Mathematical Library Mathlib [The mathlib Community(2020)].

The web version of this blueprint is available here.

1 Preliminaries

This section introduces the algebraic environment of Clifford Algebra, covering vector spaces, groups, algebras, representations, modules, multilinear algebras, quadratic forms, filtrations and graded algebras.

The material in this section should be familiar to the reader, but it is worth reading through it to become familiar with the notation and terminology that is used, as well as their counterparts in Lean, which usually require some additional treatment, both mathematically and technically (probably applicable to other formal proof verification systems).

No details will be given as these are given in standard textbooks such as [Mac Lane and Birkhoff(1999)], or see the references in corresponding section.

1.1 Basics

In this section, we follow [Jadczyk(2019)], with supplements from [Garling(2011), Chen(2016)], and modifications to match the counterparts in Lean's Mathlib.

Definition 1.1.1 (Group)**.** A **group** is a pair $(G, *)$, where G is a set, $*$ is a binary operation, satisfying:

1. $(g * h) * j = g * (h * j)$ for all $g, h, j \in G$ (**associativity**)
2. there exists e in G such that $e * g = g * e = g$ for all $g \in G$
3. for each $g \in G$ there exists $g^{-1} \in G$ such that $g * g^{-1} = g^{-1} * g = e$

Remark 1.1.2 — It then follows that e, the **identity element**, is unique, and that for each $g \in G$ the **inverse** g^{-1} is unique.

A group G is abelian, or **commutative**, if $g * h = h * g$ for all $g, h \in G$.

Remark 1.1.3 — In literatures, the binary operation are usually denoted by juxtaposition, and is understood to be a mapping $(g, h) \mapsto g * h$ from $G \times G$ to G.

Mathlib uses a slightly different way to encode this, $G \to G \to G$ is understood to be $G \to (G \to G)$, that sends $g \in G$ to a mapping that sends $h \in G$ to $g * h \in G$.

Further more, a mathimatical construct is represented by a "type", as Lean has a dependent type theory foundation.

It can be denoted multiplicatively as $*$ in Group or additively as $+$ in AddGroup, where e will be denoted by 1 or 0, respectively, sometimes with subscript (e.g. 1_G) to indicate where it is.

We will use the corresponding notation in Mathlib for future operations without further explanation.

Definition 1.1.4 (Monoid)**.** A **monoid** is a pair $(R, *)$, satisfying:

1. $(a * b) * c = a * (b * c)$ for all $a, b, c \in R$ (**associativity**),
2. There is an element $1 \in R$ such that $1 * a = a * 1 = a$ for all a in R (that is 1 is the multiplicative identity (**neutral element**).

Definition 1.1.5 (Ring)**.** A **ring** is a triple $(R, +, *)$, satisfying:

1. The elements of R form a **commutative group** under $+$;
2. The elements of R form a **monoid** under $*$;
3. If a, b, c are elements of R, we have

$$a * (b + c) = a * b + a * c,$$

$$(a + b) * c = a * c + b * c$$

(left and right **distributivity** over $+$).

Definition 1.1.6 (Division ring)**.** A ring containing at least two elements, in which every nonzero element a has a multiplicative inverse a^{-1} is called a **division ring** (sometimes also called a "skew field").

Remark 1.1.7 — In applications to Clifford algebras R will be always assumed to be **commutative**.

Definition 1.1.8 (Module)**.** Let R be a commutative ring. A **module** over R (in short R-module) is a set M such that

1. M has a structure of an additive group,
2. For every $a, b \in R$, $x, y \in M$, an operation $\alpha \bullet a$ called scalar multiplication is defined, and we have

   i $a \bullet (x + y) = a \bullet x + b \bullet y$,

   ii $(a + b) \bullet x = a \bullet x + b \bullet x$,

   iii $a * (b \bullet x) = (a * b) \bullet x$,

   iv $1_R \bullet x = x$.

Remark 1.1.9 — The notation of scalar multiplication is generalized as heterogeneous scalar multiplication in Mathlib:

$$\alpha \to \beta \to \gamma$$

where α, β, $gamma$ are different types.

Definition 1.1.10 (Vector space)**.** If R is a **division ring**, then a module M over R is called a **vector space**.

Remark 1.1.11 — For generality, Mathlib uses Module throughout for vector spaces, particularly, for a vector space V, it's usually declared as

```
variable [DivisionRing K] [AddCommGroup V] [Module K V]
```

for definitions/theorems about it, and most of them can be found under Mathlib.LinearAlgebra.

Remark 1.1.12 — A **submodule** N of M is a module N such that every element of N is also an element of M.

If M is a vector space then N is called a **subspace**.

Definition 1.1.13 (Algebra)**.** An **algebra** A over R is a module over R with a multiplication which makes A a ring and satisfying

$$\alpha(xy) = (\alpha x)y = x(\alpha y), (x, y \in A, \alpha \in R).$$

Remark 1.1.14 — What's simply called algebra is actually associative algebra with identity, a.k.a. **associative unital algebra**. See the red herring principle for more about such phenomenon for naming, particularly the example of (possibly) **nonassociative algebra**.

Definition 1.1.15 (RingHom)**.** Let $(\alpha, +_\alpha, *_\alpha)$, and $(\beta, +_\beta, *_\beta)$ be rings.

A **ring homomorphism**, from α to β is a function $f : \alpha \to \beta$ such that

(i) $f(x +_\alpha y) = f(x) +_\beta f(y)$ for each $x, y \in \alpha$.

(ii) $f(x \times_\alpha y) = f(x) \times_\beta f(y)$ for each $x, y \in \alpha$.

(iii) $f(1_\alpha) = 1_\beta$.

and is denoted as $f : \alpha \to_{+*} \beta$.

Definition 1.1.16 (FreeAlgebra)**.** TODO

Definition 1.1.17 (LinearMap)**.** TODO

Definition 1.1.18 (RingQuot)**.** TODO

Definition 1.1.19 (TensorAlgebra relation)**.** TODO

Definition 1.1.20 (Tensor algebra)**.** Let M be a module over R. An algebra T is called a **tensor algebra** over M (or "of M ") if it satisfies the following universal property

2. T is an algebra containing M as a **submodule**, and it is **generated by** M,

3. Every linear mapping λ of M into an algebra A over R, can be extended to a **homomorphism** θ of T into A.

Remark 1.1.21 — The properties above are equivalent to the following:

2. T is the free (associative, unital) R-algebra generated by M.

3. additional relations making the inclusion of M into an R-linear map

As ideals haven't been formalized for the non-commutative case, Mathlib uses RingQuot which is the quotient of a non-commutative ring by an arbitrary relation.

2 Foundations

2.1 Clifford algebras - definition

Throughout this section:
Let M be a module over a commutative ring R, equipped with a quadratic form $Q : M \to R$.
Let $\iota : M \to_{l[R]} T(M)$ be the canonical R-linear map for the tensor algebra $T(M)$.
Let $\iota_a : R \to_{+*} T(M)$ be the canonical map from R to $T(M)$, as a ring homomorphism.

Definition 2.1.1 (Clifford relation)**.** $\forall m \in M, \iota(m)^2 \sim \iota_a(Q(m))$

We say that ι **is Clifford** if this relation holds.

Definition 2.1.2 (Clifford algebra)**.** A **Clifford algebra** over M, denoted $\mathcal{C}\ell(M)$, is the quotient of the **tensor algebra** $T(M)$ by **Clifford relation** 2.1.1.

Remark 2.1.3 — In literatures, M is often written V, and the quotient is taken by the two-sided ideal I_Q generated from the set $\{v \otimes v - Q(v) \mid v \in V\}$.
As of writing, Mathlib does not have direct support for two-sided ideals, but it does support the equivalent operation of taking the quotient by a suitable closure of a relation like $v \otimes v \sim Q(v)$.
Hence the definition above.

Example 2.1.4 (Clifford algebra over a vector space)

Let V be a vector space $\mathbb{R}^n$ over $\mathbb{R}$, equipped with a quadratic form Q.
Since $\mathbb{R}$ is a commutative ring and V is a module, definition 2.1.2 of Clifford algebra applies.

2.1.1 Involutions

2.2 Structure of Clifford algebras

2.3 Classifying Clifford algebras

2.4 Representing Clifford algebras

2.5 Spin

3 Geometric Algebra

3.1 Axioms

3.2 Operations and properties

4 Concrete algebras - definition

4.1 CGA

4.2 PGA

4.3 STA

5 Applications

5.1 Geometry

References

[Chen(2016)] Evan Chen. 2016. An infinitely large napkin.

[de Moura et al.(2015)] Leonardo de Moura, Soonho Kong, Jeremy Avigad, Floris van Doorn, and Jakob von Raumer. 2015. The Lean Theorem Prover (System Description). In *Automated Deduction - CADE-25*, Amy P. Felty and Aart Middeldorp (Eds.). Lecture Notes in Computer Science, Vol. 9195. Springer International Publishing, Cham, 378–388. https://doi.org/10.1007/978-3-319-21401-6_26

[Garling(2011)] David JH Garling. 2011. *Clifford algebras: an introduction*. Vol. 78. Cambridge University Press.

[Hestenes and Sobczyk(1984)] David Hestenes and Garret Sobczyk. 1984. *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*. Vol. 5. Springer Science & Business Media. https://www.springer.com/gp/book/9789027716736

[Jadczyk(2019)] Arkadiusz Jadczyk. 2019. Notes on Clifford Algebras. (2019).

[Mac Lane and Birkhoff(1999)] Saunders Mac Lane and Garrett Birkhoff. 1999. *Algebra*. Vol. 330. American Mathematical Soc.

[Moura and Ullrich(2021)] Leonardo de Moura and Sebastian Ullrich. 2021. The lean 4 theorem prover and programming language. In *Automated Deduction–CADE 28: 28th International Conference on Automated Deduction, Virtual Event, July 12–15, 2021, Proceedings 28*. Springer, 625–635.

[The mathlib Community(2020)] The mathlib Community. 2020. The Lean Mathematical Library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs* (New Orleans, LA, USA, 2020-01-20) *(CPP 2020)*. Association for Computing Machinery, 367–381. https://doi.org/10.1145/3372885.3373824

[Ullrich(2023)] Sebastian Andreas Ullrich. 2023. *An Extensible Theorem Proving Frontend*. Ph. D. Dissertation. Dissertation, Karlsruhe, Karlsruher Institut für Technologie (KIT), 2023.