**Orthogonal and Special Orthogonal Groups in Lean Theorem Prover**

Name : Tasnia Kader

Advisor : Dr. Jaclyn Lang

May 10, 2023

Temple university

Contents

[Introduction 2](#_Toc134643487)

[Orthogonal Group 2](#_Toc134643488)

[Definition 2](#_Toc134643489)

[Instances and Coercion Lemmas 3](#_Toc134643490)

[Monoid and Group 4](#_Toc134643491)

[Linear Transformation 5](#_Toc134643492)

[Properties 5](#_Toc134643493)

[Examples 8](#_Toc134643494)

[Group Homomorphism to the Dihedral Group 8](#_Toc134643495)

[Special Orthogonal Group 9](#_Toc134643496)

[Definition 9](#_Toc134643497)

[Instances and Coercion Lemmas 9](#_Toc134643498)

[Properties & Examples 10](#_Toc134643499)

[Conclusion 10](#_Toc134643500)

# Introduction

Upon learning about Lean theorem prover, a powerful proof assistant developed at Microsoft Research, I was immediately captivated by its abilities. As someone with a lifelong passion for mathematics and computer science, the idea of a computer program that could verify proofs and even provide new insights was nothing short of intriguing. The concept of formalizing mathematics using a proof assistant was completely foreign to me, but as I delved deeper, I began to see the potential for digitizing mathematical concepts. After all, formalizing mathematics involves expressing mathematical ideas in a formal, machine-readable language using a proof assistant, which offers many benefits, including increased accuracy and the ability to verify published research work.

As I explored Lean's math library, I was surprised by the absence of theorems regarding some of the most fundamental groups: the orthogonal group and the special orthogonal group. These groups possess intriguing algebraic and geometric properties, yet they were noticeably absent from the math library. This was a major problem that needed to be addressed. Hence, my goal was to formalize the orthogonal and special orthogonal groups in Lean, contributing to the expansion of its math library and providing a basis for complex mathematical structures, proofs, and theorems.

In this paper, I will provide an overview of the orthogonal and special orthogonal groups and their formalization in Lean. I will also present the lemmas and theorems that I prove regarding these groups. I used the structure for Lean’s built-in special linear group as a guide to create the orthogonal and special orthogonal groups. The purpose was to have a sense of uniformity. I hope this implementation contributes to the expansion of Lean's math library and furthers the field of formalized mathematics.

# Orthogonal Group

## Definition

To define the orthogonal group, I needed a solid grasp of type theory, which is the foundation of Lean theorem prover. Type theory is a branch of mathematical logic that organizes mathematical objects into types based on their shared properties. Type theory is often considered as an alternative to set theory. However, there are some fundamental differences, which gave me a fresh perspective on how to view and work with various mathematical objects. Unlike set theory, where an object can belong to multiple sets, each object has only one associated type in type theory. Thus, operations and functions in type theory are only defined on objects of compatible types. Also, in set theory, new sets are formed using operations such as union, intersection, and complement. However, in Lean, commands such as inductive, structure, and abbreviation can be used to create new types. One such command is the subtype command, which is used to define the subsets of a given type.

Using this command, I define the **orthogonal group** as a subtype of m-by-m matrices with entries of type α with the following property : , where denotes the transpose of matrix . The orthogonal group consists of matrices that are a subset of all matrices restricted by the property mentioned. Thus, declaring it as a subtype is the most intuitive way to define it in Lean.

## Instances and Coercion Lemmas

To build a foundation for working with the orthogonal group in Lean, I have instantiated classes and created coercion lemmas. These tools allow me to use the group in various contexts and prove theorems about its properties and behaviors.

Instances in Lean are declarations that establish that a certain type has a particular property or attribute. For example, I instantiate the has\_coe class to declare that elements in the orthogonal group can be viewed as matrices. This allows me to access theorems related to matrices to prove propositions about the orthogonal group. This is done using subtype.val, which given an element of type orthogonal m α, returns an element of type matrix m m α using the earlier definition for an orthogonal group. The notation changes the type from orthogonal m α to matrix m m α.

To enable operations on elements of type orthogonal m α, I use coercion to view them as matrices of type matrix m m α. Moreover, to ensure that this coercion does not affect the result of operations, I have written a series of coercion lemmas that state that first coercing a variable of type orthogonal m α to matrix m m α, and then performing operations on it, is the same as first performing the operations and then coercing the result.

For example, the lemma coe\_eq\_inv\_coe states that . Thus, first taking the inverse of the variable of type orthogonal m α and coercing it to matrix m m α is the same as first coercing and then taking the inverse.

## Monoid and Group

Now, with the instances and coercion lemmas, I can prove that orthogonal matrices form a group under multiplication. First, I show that they satisfy the requirements of being a monoid.

A **monoid** is a mathematical structure that consists of a set with an associative binary operation and an identity element. In Lean, the concept of a monoid is formalized as a structure with a set of fields or axioms that must be proven to establish that a given type is indeed a monoid. The following table summarizes the fields that must be satisfied to show that a type is a monoid in Lean:

|  |  |
| --- | --- |
| **Field** | **Meaning** |
| mul | The type includes a multiplication operation |
| mul\_assoc | Multiplication is associative |
| one | The type has an identity element |
| mul\_one | Multiplying the identity element on the right returns the original element |
| one\_mul | Multiplying the identity element on the left returns the original element |

To prove that orthogonal matrices satisfy these axioms, I made use of the fact that orthogonal matrices can be coerced to matrices and thus inherit the monoid structure of the set of m-by-m matrices.

To show that the orthogonal matrices form a group, it is necessary to extend the concept of a monoid to account for the invertibility of these matrices. A **group** is a monoid with an inverse element. Thus, in addition to the fields for a monoid, a group requires two additional fields:

|  |  |
| --- | --- |
| **Field** | **Meaning** |
| inv | The type includes an inverse |
| mul\_left\_inv | Multiplication on the left by the inverse results in the original element |

To establish that orthogonal matrices satisfy these fields, I take advantage of the fact that all orthogonal matrices are invertible, with their transpose serving as the inverse. Moreover, the proposition also holds for orthogonal matrices, enabling me to prove the required fields for the group structure.

## Linear Transformation

Orthogonal matrices can be viewed as transformations that preserve distances and angles, making them an important class of linear transformations. In this section, I explore how to represent orthogonal matrices as linear transformations.

The definition to\_lin’ takes of type orthogonal m α and creates a linear map whose associated matrix is . The to\_fun field creates such a linear map while the map\_one' and map\_mul' fields of the to\_lin' structure show that the function preserves the identity and multiplication properties, respectively, under the linear map.

The next few lemmas provide the conversions between the linear maps associated with the orthogonal and matrix types in Lean.

The orthogonal group is a subgroup of the general linear group since the general linear group consists of all invertible matrices. To make this relationship more precise, I define orth\_to\_GL, which constructs a group homomorphism from the orthogonal group to the general linear group. First, I use the fact that there is a natural linear isomorphism between the general linear group and the group of invertible linear maps from a vector space to itself. This is provided by the general\_linear\_equiv function from linear\_map, which returns an isomorphism between GL α (m → α) and invertible linear maps from . Next, I consider the to\_lin’ function from earlier that creates a linear map from . I then use the mul\_equiv.to\_monoid\_hom function to convert the general\_linear\_equiv isomorphism to a group homomorphism, and finally compose it with the to\_lin' function to get a group homomorphism from the orthogonal group to the general linear group. This construction is important because it allows me to view the orthogonal group as a subgroup of the general linear group, and therefore leverage the vast array of results available for the general linear group when working with the orthogonal group.

## Properties

Not only does the orthogonal group possess intriguing algebraic properties, but also fascinating geometric implications. In this section, I explore some of these characteristics.

First, the theorem orth\_det states that the determinant of an orthogonal matrix is . This theorem is significant because it serves as a crucial building block in defining the special orthogonal group. The proof is as follows.

I begin by considering the built in theorem in lean called matrix.det\_one that states

Since is an orthogonal matrix, , so I substitute the left-hand side of this equation for to get

Now, I use matrix.det\_mul to get

The determinant of a matrix is equal to the determinant of the transpose, so

Hence, determinant of orthogonal matrices is .

Orthogonal matrices over correspond to linear transformations that preserve lengths and angles, which have a natural geometric interpretation. For the remaining properties, I consider a variable of type O(n,) where is a natural number, and the entries in the orthogonal matrix are real values.

I declare two variables and of type euclidean\_space ℝ (fin n), which can be thought of as vectors in Euclidean space of dimension n over the field of real numbers. The notation represents a function that takes an orthogonal matrix *A* and returns a linear map between Euclidean spaces.

The theorem orth\_presv\_inner establishes . In other words, the linear transformation with the associated orthogonal matrix preserves the inner product of and . I begin by applying the adjoint property of linear maps to rewrite as where is the conjugate transpose of the matrix *A*. Since the conjugate transpose is just the transpose over real numbers, I get . As is an element of the orthogonal group, , so

Therefore, , which proves that orthogonal matrices preserve the inner product of vectors in the Euclidean space.

The next theorem orth\_if\_presv\_inner is the statement that if a linear transformation preserves the inner product, it will have an associated orthogonal matrix. This theorem is slightly more challenging to prove due to the several universally quantified statements involved. My hypothesis is

where is of type matrix m m α and and are of type euclidean\_space ℝ (fin n). I use the adjoint property once again to show that the hypothesis implies

Because inner products are non-degenerate, I use ext\_inner\_right to write the statement above as

Thus, . Since I am working will real-valued entries, is . Therefore, , which is the proposition for an orthogonal matrix.

Hence, a linear transformation preserves the inner product if and only if it has an associated orthogonal matrix.

Now, I use the previous theorem to prove orth\_presv\_norm, which states that , i.e., the orthogonal matrix preserves norm of . Norm is the square root of the inner product of the vector with itself, so I rewrite as

Then, I use orth\_presv\_inner to conclude that the statement above is indeed true.

The goal of the theorem orth\_presv\_dist is to show that the linear transformation preserves the distance between and : . Using the definition of distance, I turn my goal into

I then use orth\_presv\_norm to prove the theorem.

Since isometry refers to the linear transformations that preserve distance, I use orth\_presv\_dist to prove that is an isometry.

Using orth\_presv\_inner and orth\_presv\_norm, I prove that orthogonal matrices also preserve the angle between two vectors because the formula for angle relies on the inner product and norm of vectors.

Overall, these theorems provide important insights into the behavior of elements in orthogonal groups and their effects on vectors in Euclidean space.

## Examples

In general, working with concrete examples in Lean can be more challenging than working with abstract proofs. This is because Lean is designed to handle mathematical abstractions and generalizations rather than specific examples. However, examples can provide valuable insights and understanding about a concept or theorem that may not be immediately obvious from abstract proofs, so it is crucial to include them in addition to abstract proofs.

While the theorems presented in the previous section establish that orthogonal matrices preserve norms, distances, and angles, they do not directly answer the question of what elements in this group represents. To shed light on this matter, I consider reflections, rotations, and combinations of these, which are all distance-preserving transformations of a Euclidean space, and thus are elements of the orthogonal group.

To illustrate this, I first define the two-by-two matrices for rotation, reflection, rotation followed by reflection, and reflection followed by rotation.[[1]](#footnote-1) For each proof, I find the transpose of the matrix and multiply it by the original matrix. Finally, I show that simplifying the product results in the identity matrix, which establishes that the matrix is indeed an element of O(2, ℝ).

## Group Homomorphism from the Dihedral Group

In order to further understand the symmetries and transformations of geometric objects, I now explore a notable group homomorphism. Using the fact that rotations and reflections are of type O(2, ℝ), I prove that a group homomorphism exists from the dihedral group of order n to O(2, ℝ).

The dihedral group is a group of symmetries of a regular polygon with n sides, which consists of rotations and reflections that preserve the polygon's shape. In Lean, the dihedral group is defined as an inductive type with two constructors: r and sr. I map r, which represents the rotations of a polygon, to rotation\_2D\_orth.[[2]](#footnote-2) Likewise, I map sr, which represents reflections of a polygon, to ref\_x\_rot\_orth, a combination of reflection across the x-axis and rotation.

Next, I demonstrate that this group homomorphism preserves the identity element and then proceed to show that the orthogonal group preserves multiplication in the dihedral group. By doing so, I establish the existence of a group homomorphism. This result allows for a better understanding of the relationship between the dihedral and orthogonal groups and sheds light on the symmetries and transformations of regular polygons.

# Special Orthogonal Group

The special orthogonal group is constructed in a similar manner as the orthogonal group. Therefore, in this section, I only highlight the key differences in the implementation of these groups.

## Definition

Earlier, I proved that the determinant of an orthogonal matrix is either +1 or -1. The set of elements in the orthogonal group with determinant +1 forms a subgroup of the orthogonal group, known as the **special orthogonal group**. I define this group in Lean as a subtype of orthogonal m α, with the additional property that , where is an orthogonal matrix.

## Instances and Coercion Lemmas

I once again utilize instances and coercion lemmas to establish a framework for working with this group. By defining a coercion from special\_orth m α to orthogonal m α, denoted by , I am able to apply the theorems and lemmas I proved earlier to the special orthogonal group. This allows me to use the same results for both groups, as the special orthogonal group inherits all the properties of the orthogonal group. Furthermore, because the orthogonal group has a coercion to matrices, I can also coerce elements of special\_orth m α to matrix m m α. This provides me with additional theorems to work with from the matrix library.

## Properties & Examples

The special orthogonal group, like the orthogonal group, preserves inner product, norm, distance, and angles of vectors. By coercing to the orthogonal group, I apply the theorems and lemmas previously established for the orthogonal group to prove these same properties for the special orthogonal group.

However, unlike the orthogonal group, the special orthogonal group only consists of matrices that represent rotations. For example, the two-by-two rotation matrix has the determinant which is equal to 1. Therefore, this matrix is an element of the special orthogonal group.

# Conclusion

To build a solid foundation for further research verification in Lean, it is crucial to start with the basics and formalize fundamental mathematical structures, such as the orthogonal and special orthogonal groups. Using Lean, I verified that these groups preserve the inner product, distance, angles, and norms of vectors. I demonstrated that rotations and reflections are elements of the orthogonal group, while only rotations describe the special orthogonal group. This interplay between algebra and geometry allowed me to gain a deeper understanding of these groups. In addition, I explored an important group homomorphism from the dihedral group to the orthogonal group in two dimensions, which provided insights into the behavior of both.

The orthogonal and special orthogonal groups play a significant role in group theory, linear algebra, and geometry, so I hope their implementation in Lean will provide a strong framework for verifying various mathematical concepts and results in the future.

1. These matrices can be found in the auxiliary\_theorems.lean file. [↑](#footnote-ref-1)
2. The proof can be found in the orth\_to\_dihedral.lean file. [↑](#footnote-ref-2)