A FREE GROUP OF ROTATIONS OF RANK 2

Jagadish Bapanapally & Ruben Gamboa

University of Wyoming

ACL2-2022

Summary

- Define a set of 3D matrices
- Prove all the elements of the set are different from each other
- Formalize 3D rotations in ACL2(r)
- Prove every element of the set is a rotation

[1] Weston, T.. “THE BANACH-TARSKI PARADOX.” (2003).

Motivation

- The Banach-Tarski theorem in ACL2(r)
- "The Banach-Tarski Paradox" by Tom Weston[1]

[1] Weston, T.. "THE BANACH-TARSKI PARADOX." (2003).

Method

- Define a set of words of rank 2
  example words: aa, bb, $a^{-1}b$, …
- Prove group properties for the set of words
- Associate a, b, a^{-1}, b^{-1} with specific 3D rotations
  For example:

$$A^{\pm} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{3} & \mp\frac{2\sqrt{2}}{3} \\ 0 & \pm\frac{2\sqrt{2}}{3} & \frac{1}{3} \end{bmatrix} \quad B^{\pm} = \begin{bmatrix} \frac{1}{3} & \mp\frac{2\sqrt{2}}{3} & 0 \\ \pm\frac{2\sqrt{2}}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Compose each word with matrix multiplication
  The 3D matrix we get with the word *aa* is A×A
- Prove these matrices are rotations and different from each other

Outline

1. A Free Group of Reduced Words
2. A Free Group of 3D matrices
3. A Free Group of 3D Rotations of rank 2
4. Next steps

Outline

1. A Free Group of Reduced Words
2. A Free Group of 3D matrices
3. A Free Group of 3D Rotations of rank 2
4. Next steps

Weak Word

(weak-wordp w) returns true if w is:

- An Empty list, or
- A list of characters that contains a or a^{-1} or b or b^{-1}
  - e.g., (), (a b b^{-1})
- In the ACL2(r) source files we have used characters #\a, #\b, #\c and #\d respectively

Reduced Word

(reducedwordp w) returns true if w is:

- a weak-word, and
- a and a^{-1} or b and b^{-1} does not appear next to each other in the list
  - e.g., (), (a b b a^{-1})
  - (a a^{-1}) is not a reduced word

A reduced word $\Rightarrow$ it is a weak word

Compose - Group operation

(compose x y) = (word-fix (append x y))

where x and y are weak-words

- **append** operation appends two lists
- **word-fix** fixes the list so that the result is a reduced

  word

  - e.g., (compose (a b b a) (a^{-1} b^{-1} b))

    = (word-fix (a b b a a^{-1} b^{-1} b))

    = (a b b)

Useful Lemmas

- (reducedwordp x) $\Rightarrow$ (weak-wordp x)
- (weak-wordp x) $\Rightarrow$ (reducedwordp (word-fix x))
- (weak-wordp x) $\wedge$ ((word-fix x) = x)

  $\Rightarrow$ (reducedwordp x)
- (weak-wordp x) $\Rightarrow$ (weak-wordp (cdr x))
- (reducedwordp x) $\Rightarrow$ ((word-fix x) = x)
- (reducedwordp x) $\Rightarrow$ (reducedwordp (cdr x))

Inverse Operation

(word-inverse x) = (rev (word-flip x))

where x is a weak-word

- **word-flip** operation changes the characters a to a^{-1}, b to b^{-1}, a^{-1} to a and b^{-1} to b.
  - e.g., (word-flip (a b b)) = (a^{-1} b^{-1} b^{-1})

The Identity Element

- Empty list is the identity element
- (reducedwordp x) $\Rightarrow$ (word-fix (append () x)) = x

  $\Rightarrow$ (compose () x) = x
- (reducedwordp x) $\Rightarrow$ (word-fix (append x ())) = x

  $\Rightarrow$ (compose x ()) = x

Closure Property

Intermediate Lemmas:

- (weak-wordp x) $\wedge$ (weak-wordp y)

  $\Rightarrow$ (weak-wordp (append x y))

- (reducedwordp x) $\wedge$ (reducedwordp y)

  $\Rightarrow$ (weak-wordp (append x y))

Closure:

(reducedwordp x) $\wedge$ (reducedwordp y)

$\Rightarrow$ (reducedwordp (word-fix (append x y)))

[$\because$ (weak-wordp x) $\Rightarrow$ (reducedwordp (word-fix x))]

$\Rightarrow$ (reducedwordp (compose x y))

Associative Property

Key Lemmas:

- (weak-wordp x) ∧ (weak-wordp y) ∧ (weak-wordp z)

  ⇒ (word-fix (append x y z))

  = (word-fix (append x (word-fix (append y z))))
- (weak-wordp x) ⇒ (word-fix (rev x)) = (rev (word-fix x))
  - Proved by induction on x
  - e.g., (word-fix (rev (a b b^{-1})) = (word-fix (b^{-1} b a)) = (a)

    (rev (word-fix (a b b^{-1})) = (rev (a)) = (a)

Associative Property

(reducedwordp x) $\wedge$ (reducedwordp y) $\wedge$ (reducedwordp z) $\Rightarrow$

(compose (compose x y) z) = (compose x (compose y z))

Inverse Property

Intermediate Lemmas:

- (reducedwordp x) $\Rightarrow$ (reducedwordp (word-inverse x))
  - (word-fix (rev x)) = (rev x)
  - (reducedwordp (word-flip x))
- (weak-wordp x) $\Rightarrow$ (word-inverse (word-inverse x)) = x
  - n^{th} values of (word-inverse (word-inverse x) and x are equal
  - Using equal-by-nths lemma

Inverse exists (using the above lemmas and the associativity) **:**

- (reducedwordp x) $\Rightarrow$ (compose x (word-inverse x)) = ()
- (reducedwordp x) $\Rightarrow$ (compose (word-inverse x) x) = ()

Outline

1. A Free Group of Reduced Words
2. A Free Group of 3D matrices
3. A Free Group of 3D Rotations of rank 2
4. Next steps

Matrix Algebra

1. Gamboa, Ruben, John Cowles, and J. V. Baalen. "Using ACL2 arrays to formalize matrix algebra." *Fourth International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2'03)*. Vol. 1. 2003.

2. Lambán, Laureano, Francisco J. Martín-Mateos, Julio Rubio, and José-Luis Ruiz-Reina. "Using abstract stobjs in ACL2 to compute matrix normal forms." In *International Conference on Interactive Theorem Proving*, pp. 354-370. Springer, Cham, 2017.

Matrix Algebra using array2p

- matrix equivalence
- matrix multiplication
- matrix transpose
- scalar multiplication
- Associativity of the matrix multiplication
- properties about dimensions of the matrices

Newly added:

  - r3-matrixp (predicate for a 3D matrix in ACL2(r))
  - r3-m-determinant (determinant of a 3D matrix)
  - r3-m-inverse (Inverse of a 3D matrix)

UW

A Set of 3D Matrices

$$A^{\pm} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{3} & \mp\frac{2\sqrt{2}}{3} \\ 0 & \pm\frac{2\sqrt{2}}{3} & \frac{1}{3} \end{bmatrix} \quad B^{\pm} = \begin{bmatrix} \frac{1}{3} & \mp\frac{2\sqrt{2}}{3} & 0 \\ \pm\frac{2\sqrt{2}}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(rotation w) ≔ if w is an empty list then return *id-matrix*

else if (car w) = a: return (A × (rotation (cdr w))

else if (car w) = b: return (B × (rotation (cdr w)))

else if (car w) = a^{-1}: return (A^{-1} × (rotation (cdr w)))

else if (car w) = b^{-1}: return (B^{-1} × (rotation (cdr w)))

set = { (rotation w) | (reducedwordp w) }

A Free Group of 3D matrices

Let $p = (0,1,0)$, then:

If (reducedwordp w) $\wedge$ ((len w) = n) $\wedge$ (((rotation w) $\times$ p) = q) $\wedge$ $n > 0$

we have shown by induction:

q is of the form $(\frac{1}{3}^n)(a\sqrt{2}, b, c\sqrt{2})$

where a, b and c are integers

If w is an empty list, then (rotation w) is an identity matrix, then

$q = p = (0,1,0) \Rightarrow a \equiv b \equiv c \equiv 0 \pmod{3}$

(This is not possible)

UW

A Free Group of 3D matrices

If w is a reduced word and let's say (n-mod3 w) = (a,b,c) (mod 3), then

(n-mod3 *a*w) = (0, b - c, c - b) (mod 3)
(n-mod3 a^{-1}w) = (0, b + c, c + b) (mod 3)
(n-mod3 *b*w) = (a + b, a + b, 0) (mod 3)
(n-mod3 b^{-1}w) = (a - b, b - a, 0) (mod 3)

By induction:

If w is a reduced word, and

- if (car w) = *a*, then **(n-mod3 w)** = (0, 1, 2) or (0, 2, 1)
- if (car w) = a^{-1}, then **(n-mod3 w)** = (0, 1, 1) or (0, 2, 2)
- if (car w) = *b*, then **(n-mod3 w)** = (1, 1, 0) or (2, 2, 0)
- if (car w) = b^{-1}, then **(n-mod3 w)** = (2, 1, 0) or (1, 2, 0)

∴ **(n-mod3 w)** ≠ (0,0,0) ∧ ((len w) > 0) ⇒ (rotation w) ≠ *I*

UW

A Free Group of 3D Matrices

Key lemma:

if w_1, w_2 ,are reduced words then

(rotation w_1) × (rotation w_2) = (rotation (compose w_1 w_2)

Since (rotation w) ≠ I, using the above lemma:

if w_1, w_2 ,are two different reduced words having length > 1 then

(rotation w_1) ≠ (rotation w_2)

UW

Modular addition and subtraction

(A + B) mod C = (A mod C + B mod C) mod C

(A - B) mod C = (A mod C - B mod C) mod C

Proved using properties from the book:

workshops/1999/embedded/Exercises/Exercise1-2/Exercise1.2

Outline

1. A Free Group of Reduced Words
2. A Free Group of 3D matrices
3. A Free Group of 3D Rotations of rank 2
4. Next steps

3D Rotations

M is a 3D rotation if:

- M is a 3D matrix
- $M^{-1} = M^T$
- $\det(M) = 1$

Properties of 3D Rotations

1. (r3-matrixp m_1) $\wedge$ (r3-matrixp m_2) $\Rightarrow$ (r3-matrix ($m_1 \times m_2$))
2. (r3-matrixp m_1) $\wedge$ (r3-matrixp m_2) $\Rightarrow$ $\det(m_1 \times m_2)) = \det(m_1) \times \det(m_2)$
3. (r3-matrixp m_1) $\wedge$ (r3-matrixp m_2) $\wedge$ $\det(m_1) \neq 0 \wedge \det(m_2) \neq 0$

   $(m_1 \times m_2)^{-1} = m_2^{-1} \times m_1^{-1}$
4. (r3-rotationp m) $\Rightarrow$ (r3-rotationp m^{-1})
5. (r3-rotationp m_1) $\wedge$ (r3-rotationp m_2) $\Rightarrow$ (r3-rotationp ($m_1 \times m_2$))
6. Rotations preserve distance

   If $p_1 = (x_1, y_1, z_1)$ and $p_2 = R \times p1 = (x_2, y_2, z_2)$, then

   $x_1^2 + y_1^2 + z_1^2 = x_2^2 + y_2^2 + z_2^2$

UW

A Free Group of Rotations

By induction, if w is a reduced word, then by using the properties:

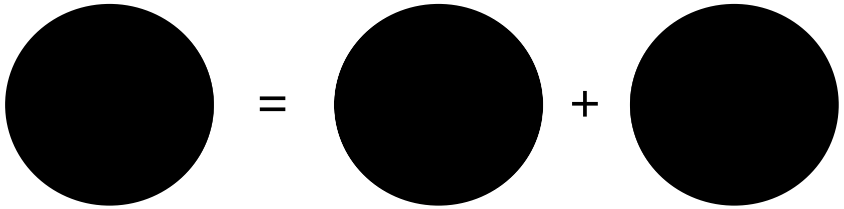
- (rotation w) is a 3D rotation

Outline

1. A Free Group of Reduced Words
2. A Free Group of 3D matrices
3. A Free Group of 3D Rotations of rank 2
4. Next steps

The Banach-Tarski Theorem

Any Solid ball B in R^3 can be cut into finitely many pieces which can be rotated to form 2 copies of B[1]



[1] Weston, T.. “THE BANACH-TARSKI PARADOX.” (2003).

The Hausdorff Paradox

There is a countable set $D \subseteq S^2$ such that

$S^2 - D$ can be divided into 5 pieces which can be rotated to form 2 copies of $S^2 - D$.[1]

[1] Weston, T.. "THE BANACH-TARSKI PARADOX." (2003).

The Hausdorff Paradox

For the set of reduced words $W(a,b)$:

- $W(a,b) = () \cup W(a) \cup W(a^{-1}) \cup W(b) \cup W(b^{-1})$
- $W(a,b) = a^{-1}W(a) \cup W(a^{-1})$
- $W(a,b) = b^{-1}W(b) \cup W(b^{-1})$

If $R(a,b)$ is the free group, an orbit of a point p on $S^2 = \{ \rho(p) \mid \rho \in R(a,b)\}$, then

- $S^2\text{-}D = R(a,b)C = C \cup R(a)C \cup R(a^{-1})C \cup R(b)C \cup R(b^{-1})C$
- $S^2\text{-}D = A^{-1}(R(a)C) \cup R(a^{-1})C$
- $S^2\text{-}D = B^{-1}(R(b)C) \cup R(b^{-1})C$

where C is the choice set

References

- Weston, T.. "THE BANACH-TARSKI PARADOX." (2003).
- Gamboa, Ruben, John Cowles, and J. V. Baalen. "Using ACL2 arrays to formalize matrix algebra." *Fourth International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2'03)*. Vol. 1. 2003.
- Bertoli, Piergiorgio, and Paolo Traverso. "Design verification of a safety-critical embedded verifier." *Computer-Aided Reasoning*. Springer, Boston, MA, 2000. 233-245.
- Madeline Tremblay (2017). *The Banach-Tarski Paradox*. Mathematics Department, University of Connecticut, Hartford, CT, USA.
- https://en.wikipedia.org/wiki/Rotation_matrix

Thank You