

Hausdorff Paradox in ACL2(r)

Definitions and Theorems

Jagadish Bapanapally

University of Wyoming
Laramie, Wyoming
jbapanap@uwyo.edu

1 Introduction

This is a summary of the Hausdorff paradox with all the definitions and theorems.

Statement: 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. Definition of a group of reduced words in ACL2(r):

Representing in Math:

$$F(a, b) = \{1\} \uplus w(a) \uplus w(a^{-1}) \uplus w(b) \uplus w(b^{-1})$$

```
1 (defun reducedwordp (x)
2   (or (a-wordp x)
3       (a-inv-wordp x)
4       (b-wordp x)
5       (b-inv-wordp x)
6       (equal x '())))
```

I have proved that the sets a-wordp, a-inv-wordp, b-wordp, b-inv-wordp and '() are disjoint in ACL2(r).

corollaries:

$$F(a, b) = a^{-1}w(a) \uplus w(a^{-1})$$

$$F(a, b) = b^{-1}w(b) \uplus w(b^{-1})$$

Definitions in ACL2(r):

$a^{-1}w(a)$:

```
1 (defun-sk a-inv*w-a-p (w)
2   (exists word-a
3     (and (a-wordp word-a)
4          (equal (compose (list (wa-inv)) word-a) w))))
```

$b^{-1}w(b)$:

```
5 (defun-sk b-inv*w-b-p (w)
6   (exists word-b
7     (and (b-wordp word-b)
8          (equal (compose (list (wb-inv)) word-b) w))))
```

Submitted to:

corollaries in ACL2(r):

```

9 (defthmd reducedword-equiv-4
10   (iff (reducedwordp w)
11         (or (a-inv*w-a-p w)
12             (a-inv-wordp w))))

13 (defthmd reducedword-equiv-2
14   (iff (reducedwordp w)
15         (or (b-inv*w-b-p w)
16             (b-inv-wordp w))))

```

2. Definition of a point in \mathbb{R}^3 in ACL2(r)

```

17 (defun point-in-r3 (x)
18   (and (array2p :fake-name x)
19        (equal (car (dimensions :fake-name x)) 3)
20        (equal (cadr (dimensions :fake-name x)) 1)
21        (realp (aref2 :fake-name x 0 0))
22        (realp (aref2 :fake-name x 1 0))
23        (realp (aref2 :fake-name x 2 0))))

```

3. Set of points that belong to the surface of a unit sphere centered at the origin

```

24 (defun s2-def-p (point)
25   (and (point-in-r3 point)
26        (equal (+ (* (aref2 :fake-name point 0 0) (aref2 :fake-name point 0 0))
27                  (* (aref2 :fake-name point 1 0) (aref2 :fake-name point 1 0))
28                  (* (aref2 :fake-name point 2 0) (aref2 :fake-name point 2 0)))
29          1)))

```

Let's say this set is S^2 .

There is a one to one relation between the set of rotations and the set of reduced words. Lets name the set of rotations as $R(a,b)$. Then,

$$R(a,b) = 1 \uplus R(a) \uplus R(a^{-1}) \uplus R(b) \uplus R(b^{-1})$$

4. Set of all points of S^2 fixed by any non-identity rotation $\rho \in R(a,b)$

In math: $D = \{p \in S^2 \mid \text{There is a } \rho \in R(a,b), \rho \neq 1, \text{ with } \rho(p) = p\}$

Definition in ACL2(r):

```

30 (defun-sk word-exists (point)
31   (exists w
32     (and (reducedwordp w)
33           w
34           (m-= (m-* (rotation w (acl2-sqrt 2)) point)
35                point))))

36 (defun d-p (point)
37   (and (s2-def-p point)
38        (word-exists point)))

```

5. $S^2 - D$ in ACL2(r):**Definition in ACL2(r):**

```

39 (defun s2-d-p (point)
40   (and (s2-def-p point)
41        (not (d-p point))))

```

6. **Orbit of a point** $p \in S^2$

In math: $\{\rho(p) \mid \rho \in R(a,b)\}$

Definition in ACL2(r):

Returns true if o-point belongs to the orbit of point.

```

42 (defun-sk orbit-point-p-q (o-point point)
43   (exists w
44     (and (reducedwordp w)
45           (m-= (m-* (rotation w (acl2-sqrt 2)) point) o-point))))

```

7. **Choice set for the orbits of points in the set** $S^2 - D$:

Lets name this set M .

Definition in ACL2(r):

```

46 (defchoose choice-set-s2-d-p (c-point) (p)
47   (and (point-in-r3 c-point)
48        (orbit-point-p-q c-point p))
49   :strengthen t)

```

8. **Partitions of** $S^2 - D$

(a) First partition:

In math:

$$S^2 - D = R(a,b)M = M \uplus R(a)M \uplus R(a^{-1})M \uplus R(b)M \uplus R(b^{-1})M$$

Definitions in ACL2(r):

Set $R(a,b)M$:

```

50 (defun-sk diff-s2-d-p-q-1 (cp1 p)
51   (exists w
52     (and (reducedwordp w)
53           (m-= (m-* (rotation w (acl2-sqrt 2)) cp1) p))))
54
55 (defun-sk diff-s2-d-p-q (p)
56   (exists p1
57     (and (s2-d-p p1)
58           (diff-s2-d-p-q-1 (choice-set-s2-d-p p1) p))))
59
60 (defun diff-s2-d-p (p)
61   (and (point-in-r3 p)
62        (diff-s2-d-p-q p))

```

Set M :

```

63 (defun-sk diff-n-s2-d-p-q-1 (cp1 p)
64   (exists w
65     (and (reducedwordp w)
66           (equal w nil)
67           (m-= (m-* (rotation w (acl2-sqrt 2)) cp1) p))))
68
69 (defun-sk diff-n-s2-d-p-q (p)
70   (exists p1
71     (and (s2-d-p p1)
72           (diff-n-s2-d-p-q-1 (choice-set-s2-d-p p1) p))))
73
74 (defun diff-n-s2-d-p (p)
75   (and (point-in-r3 p)
76         (diff-n-s2-d-p-q p)))

```

Set $R(a)M$:

```

78 (defun-sk diff-a-s2-d-p-q-1 (cp1 p)
79   (exists w
80     (and (a-wordp w)
81           (m-= (m-* (rotation w (acl2-sqrt 2)) cp1) p))))
82
83 (defun-sk diff-a-s2-d-p-q (p)
84   (exists p1
85     (and (s2-d-p p1)
86           (diff-a-s2-d-p-q-1 (choice-set-s2-d-p p1) p))))
87
88 (defun diff-a-s2-d-p (p)
89   (and (point-in-r3 p)
90         (diff-a-s2-d-p-q p)))

```

Set $R(b)M$:

```

91 (defun-sk diff-b-s2-d-p-q-1 (cp1 p)
92   (exists w
93     (and (b-wordp w)
94           (m-= (m-* (rotation w (acl2-sqrt 2)) cp1) p))))
95
96 (defun-sk diff-b-s2-d-p-q (p)
97   (exists p1
98     (and (s2-d-p p1)
99           (diff-b-s2-d-p-q-1 (choice-set-s2-d-p p1) p))))
100
101 (defun diff-b-s2-d-p (p)
102   (and (point-in-r3 p)
103         (diff-b-s2-d-p-q p)))

```

Set $R(a^{-1})M$:

```

104 (defun-sk diff-a-inv-s2-d-p-q-1 (cp1 p)
105   (exists w
106     (and (a-inv-wordp w)
107           (m-= (m-* (rotation w (acl2-sqrt 2)) cp1) p))))

```

```

108
109 (defun-sk diff-a-inv-s2-d-p-q (p)
110   (exists p1
111     (and (s2-d-p p1)
112           (diff-a-inv-s2-d-p-q-1 (choice-set-s2-d-p p1) p))))
113
114 (defun diff-a-inv-s2-d-p (p)
115   (and (point-in-r3 p)
116         (diff-a-inv-s2-d-p-q p)))

```

Set $R(b^{-1})M$:

```

117 (defun-sk diff-b-inv-s2-d-p-q-1 (cp1 p)
118   (exists w
119     (and (b-inv-wordp w)
120           (m-= (m-* (rotation w (acl2-sqrt 2)) cp1) p))))
121
122 (defun-sk diff-b-inv-s2-d-p-q (p)
123   (exists p1
124     (and (s2-d-p p1)
125           (diff-b-inv-s2-d-p-q-1 (choice-set-s2-d-p p1) p))))
126
127 (defun diff-b-inv-s2-d-p (p)
128   (and (point-in-r3 p)
129         (diff-b-inv-s2-d-p-q p)))

```

Theorems in ACL2(r):

```

130 (defthmd s2-d-p-equiv
131   (iff (s2-d-p p)
132         (diff-s2-d-p p)))
133
134 (defthmd diff-s2-d-p-equivalence-1
135   (iff (diff-s2-d-p p)
136         (or (diff-n-s2-d-p p)
137             (diff-a-s2-d-p p)
138             (diff-a-inv-s2-d-p p)
139             (diff-b-s2-d-p p)
140             (diff-b-inv-s2-d-p p))))
141
142 (defthmd s2-d-p-equivalence-1
143   (iff (s2-d-p p)
144         (or (diff-n-s2-d-p p)
145             (diff-a-s2-d-p p)
146             (diff-a-inv-s2-d-p p)
147             (diff-b-s2-d-p p)
148             (diff-b-inv-s2-d-p p))))

```

(b) Second partition

In math:

$$S^2 - D = a^{-1}(R(a)M) \uplus R(a^{-1})M$$

Definitions in ACL2(r):Set $a^{-1}(R(a)M)$:

```

149 (defun-sk diff-a-inv-wa-s2-d-p-q-1 (cp1 p)
150   (exists w
151     (and (a-inv*w-a-p w)
152           (m-= (m-* (rotation w (acl2-sqrt 2)) cp1) p))))
153
154
155 (defun-sk diff-a-inv-wa-s2-d-p-q (p)
156   (exists p1
157     (and (s2-d-p p1)
158           (diff-a-inv-wa-s2-d-p-q-1 (choice-set-s2-d-p p1) p))))
159
160 (defun diff-a-inv-wa-s2-d-p (p)
161   (and (point-in-r3 p)
162         (diff-a-inv-wa-s2-d-p-q p)))

```

Set $a^{-1}(R(a)M)$:

```

163 (defun-sk a-inv-diff-a-s2-d-p-1 (p)
164   (exists p1
165     (and (diff-a-s2-d-p p1)
166           (m-= (m-* (rotation (list (wa-inv)) (acl2-sqrt 2)) p1) p))))
167
168 (defun a-inv-diff-a-s2-d-p (p)
169   (and (point-in-r3 p)
170         (a-inv-diff-a-s2-d-p-1 p)))

```

Theorems in ACL2(r):

```

171 (defthmd diff-a-inv-wa-s2-d-p-equiv
172   (iff (diff-a-inv-wa-s2-d-p p)
173         (a-inv-diff-a-s2-d-p p)))
174
175 (defthmd diff-s2-d-p-equivalence-2
176   (iff (diff-s2-d-p p)
177         (or (diff-a-inv-wa-s2-d-p p)
178             (diff-a-inv-s2-d-p p))))
179
180 (defthmd s2-d-p-equivalence-2
181   (iff (s2-d-p p)
182         (or (a-inv-diff-a-s2-d-p p)
183             (diff-a-inv-s2-d-p p))))

```

(c) Third partition

In math:

$$S^2 - D = b^{-1}(R(b)M) \uplus R(b^{-1})M$$

Definitions in ACL2(r):Set $b^{-1}(R(b)M)$:

```

184 (defun-sk diff-b-inv-wb-s2-d-p-q-1 (cp1 p)
185   (exists w

```

```

186      (and (b-inv*w-b-p w)
187            (m-= (m-* (rotation w (acl2-sqrt 2)) cp1) p))))
188
189 (defun-sk diff-b-inv-wb-s2-d-p-q (p)
190   (exists p1
191     (and (s2-d-p p1)
192           (diff-b-inv-wb-s2-d-p-q-1 (choice-set-s2-d-p p1) p))))
193
194 (defun diff-b-inv-wb-s2-d-p (p)
195   (and (point-in-r3 p)
196         (diff-b-inv-wb-s2-d-p-q p)))

```

Set $b^{-1}(R(b)M)$:

```

197 (defun-sk b-inv-diff-b-s2-d-p-1 (p)
198   (exists p1
199     (and (diff-b-s2-d-p p1)
200           (m-= (m-* (rotation (list (wb-inv)) (acl2-sqrt 2)) p1) p))))
201
202 (defun b-inv-diff-b-s2-d-p (p)
203   (and (point-in-r3 p)
204         (b-inv-diff-b-s2-d-p-1 p)))

```

Theorems in ACL2(r):

```

205 (defthmd diff-b-inv-wb-s2-d-p-equiv
206   (iff (diff-b-inv-wb-s2-d-p p)
207         (b-inv-diff-b-s2-d-p p)))
208
209 (defthmd diff-s2-d-p-equivalence-3
210   (iff (diff-s2-d-p p)
211         (or (diff-b-inv-wb-s2-d-p p)
212             (diff-b-inv-s2-d-p p))))
213
214 (defthmd s2-d-p-equivalence-3
215   (iff (s2-d-p p)
216         (or (b-inv-diff-b-s2-d-p p)
217             (diff-b-inv-s2-d-p p))))

```

I have proved that the sets $(\text{diff-n-s2-d-p } p)$, $(\text{diff-a-s2-d-p } p)$, $(\text{diff-a-inv-s2-d-p } p)$, $(\text{diff-b-s2-d-p } p)$, $(\text{diff-b-inv-s2-d-p } p)$ are disjoint and the sets $(\text{a-inv-diff-a-s2-d-p } p)$, $(\text{diff-a-inv-s2-d-p } p)$ are disjoint and the sets $(\text{b-inv-diff-b-s2-d-p } p)$, $(\text{diff-b-inv-s2-d-p } p)$ are disjoint.

2 To Do

To completely prove the Hausdorff paradox, I have to prove that the set D is countable.