

The independence of Tarski’s Euclidean axiom

T. J. M. Makarios

February 6, 2026

Abstract

Tarski’s axioms of plane geometry are formalized and, using the standard real Cartesian model, shown to be consistent. A substantial theory of the projective plane is developed. Building on this theory, the Klein–Beltrami model of the hyperbolic plane is defined and shown to satisfy all of Tarski’s axioms except his Euclidean axiom; thus Tarski’s Euclidean axiom is shown to be independent of his other axioms of plane geometry.

An earlier version of this work was the subject of the author’s MSc thesis [2], which contains natural-language explanations of some of the more interesting proofs.

Contents

1	Metric and semimetric spaces	2
2	Miscellaneous results	3
3	Tarski’s geometry	8
3.1	The axioms	8
3.2	Semimetric spaces satisfy the first three axioms	9
3.3	Some consequences of the first three axioms	9
3.4	Some consequences of the first five axioms	11
3.5	Simple theorems about betweenness	11
3.6	Simple theorems about congruence and betweenness	12
4	Real Euclidean space and Tarski’s axioms	13
4.1	Real Euclidean space satisfies the first five axioms	13
4.2	Real Euclidean space also satisfies axioms 6, 7, and 11	13
4.3	Real Euclidean space satisfies the Euclidean axiom	13
4.4	The real Euclidean plane	13
4.5	Special cases of theorems of Tarski’s geometry	14
5	Linear algebra	14
5.1	Matrices	16

6	Right group actions	17
7	Projective geometry	18
7.1	Proportionality on non-zero vectors	18
7.2	Points of the real projective plane	19
7.3	Lines of the real projective plane	20
7.4	Collineations of the real projective plane	25
7.4.1	As a group	27
7.4.2	As a group action	28
7.4.3	Parts of some Statements from [1]	30
7.5	Cross ratios	31
7.6	Cartesian subspace of the real projective plane	33
8	The hyperbolic plane and Tarski's axioms	36
8.1	Characterizing a specific conic in the projective plane	36
8.2	Some specific points and lines of the projective plane	40
8.3	Definition of the Klein–Beltrami model of the hyperbolic plane	42
8.4	K -isometries map the interior of the conic to itself	45
8.5	The K -isometries form a group action	49
8.6	The Klein–Beltrami model satisfies Tarski's first three axioms	49
8.7	Some lemmas about betweenness	50
8.8	The Klein–Beltrami model satisfies axiom 4	52
8.9	More betweenness theorems	53
8.10	Perpendicularity	55
8.11	Functions of distance	58
8.11.1	A formula for a cross ratio involving a perpendicular foot	62
8.12	The Klein–Beltrami model satisfies axiom 5	62
8.13	The Klein–Beltrami model satisfies axioms 6, 7, and 11	63
8.14	The Klein–Beltrami model satisfies the dimension-specific ax- ioms	63
8.15	The Klein–Beltrami model violates the Euclidean axiom	64

1 Metric and semimetric spaces

```

theory Metric
imports HOL-Analysis.Multivariate-Analysis
begin

locale semimetric =
  fixes dist :: 'p ⇒ 'p ⇒ real
  assumes nonneg [simp]: dist x y ≥ 0
  and eq-0 [simp]: dist x y = 0 ⟷ x = y
  and symm: dist x y = dist y x
begin

```

lemma *refl [simp]: dist x x = 0*
 ⟨*proof*⟩
end

locale *metric =*
fixes *dist :: 'p ⇒ 'p ⇒ real*
assumes *[simp]: dist x y = 0 ⟷ x = y*
and *triangle: dist x z ≤ dist y x + dist y z*

sublocale *metric < semimetric*
 ⟨*proof*⟩

definition *norm-dist :: ('a::real-normed-vector) ⇒ 'a ⇒ real* **where**
[simp]: norm-dist x y ≜ norm (x - y)

interpretation *norm-metric: metric norm-dist*
 ⟨*proof*⟩

end

2 Miscellaneous results

theory *Miscellany*
imports *Metric*
begin

lemma *unordered-pair-element-equality:*
assumes $\{p, q\} = \{r, s\}$ **and** $p = r$
shows $q = s$
 ⟨*proof*⟩

lemma *unordered-pair-equality:* $\{p, q\} = \{q, p\}$
 ⟨*proof*⟩

lemma *cosine-rule:*
fixes $a\ b\ c :: \text{real} \wedge ('n::\text{finite})$
shows $(\text{norm-dist } a\ c)^2 =$
 $(\text{norm-dist } a\ b)^2 + (\text{norm-dist } b\ c)^2 + 2 * ((a - b) \cdot (b - c))$
 ⟨*proof*⟩

lemma *scalar-equiv:* $r * s\ x = r *_R\ x$
 ⟨*proof*⟩

lemma *norm-dist-dot:* $(\text{norm-dist } x\ y)^2 = (x - y) \cdot (x - y)$
 ⟨*proof*⟩

definition *dep2 :: 'a::real-vector ⇒ 'a ⇒ bool* **where**
*dep2 u v ≜ ∃ w r s. u = r *_R w ∧ v = s *_R w*

lemma *real2-eq*:

fixes $u\ v :: \text{real}^2$

assumes $u\$1 = v\1 **and** $u\$2 = v\2

shows $u = v$

<proof>

definition *rotate2* :: $\text{real}^2 \Rightarrow \text{real}^2$ **where**

$\text{rotate2}\ x \triangleq \text{vector}\ [-x\$2, x\$1]$

declare *vector-2* [*simp*]

lemma *rotate2* [*simp*]:

$(\text{rotate2}\ x)\$1 = -x\2

$(\text{rotate2}\ x)\$2 = x\1

<proof>

lemma *rotate2-rotate2* [*simp*]: $\text{rotate2}\ (\text{rotate2}\ x) = -x$

<proof>

lemma *rotate2-dot* [*simp*]: $(\text{rotate2}\ u) \cdot (\text{rotate2}\ v) = u \cdot v$

<proof>

lemma *rotate2-scaleR* [*simp*]: $\text{rotate2}\ (k *_R x) = k *_R (\text{rotate2}\ x)$

<proof>

lemma *rotate2-uminus* [*simp*]: $\text{rotate2}\ (-x) = -(\text{rotate2}\ x)$

<proof>

lemma *rotate2-eq* [*iff*]: $\text{rotate2}\ x = \text{rotate2}\ y \longleftrightarrow x = y$

<proof>

lemma *dot2-rearrange-1*:

fixes $u\ x :: \text{real}^2$

assumes $u \cdot x = 0$ **and** $x\$1 \neq 0$

shows $u = (u\$2 / x\$1) *_R (\text{rotate2}\ x)$ (**is** $u = ?u'$)

<proof>

lemma *dot2-rearrange-2*:

fixes $u\ x :: \text{real}^2$

assumes $u \cdot x = 0$ **and** $x\$2 \neq 0$

shows $u = -(u\$1 / x\$2) *_R (\text{rotate2}\ x)$ (**is** $u = ?u'$)

<proof>

lemma *dot2-rearrange*:

fixes $u\ x :: \text{real}^2$

assumes $u \cdot x = 0$ **and** $x \neq 0$

shows $\exists k. u = k *_R (\text{rotate2}\ x)$

<proof>

lemma *real2-orthogonal-dep2*:
fixes $u\ v\ x :: \text{real}^2$
assumes $x \neq 0$ **and** $u \cdot x = 0$ **and** $v \cdot x = 0$
shows $\text{dep2}\ u\ v$
 $\langle \text{proof} \rangle$

lemma *dot-left-diff-distrib*:
fixes $u\ v\ x :: \text{real}^n$
shows $(u - v) \cdot x = (u \cdot x) - (v \cdot x)$
 $\langle \text{proof} \rangle$

lemma *dot-right-diff-distrib*:
fixes $u\ v\ x :: \text{real}^n$
shows $x \cdot (u - v) = (x \cdot u) - (x \cdot v)$
 $\langle \text{proof} \rangle$

lemma *am-gm2*:
fixes $a\ b :: \text{real}$
assumes $a \geq 0$ **and** $b \geq 0$
shows $\text{sqrt}\ (a * b) \leq (a + b) / 2$
and $\text{sqrt}\ (a * b) = (a + b) / 2 \iff a = b$
 $\langle \text{proof} \rangle$

lemma *refl-on-allrel*: $\text{refl-on}\ A\ (A \times A)$
 $\langle \text{proof} \rangle$

lemma *refl-on-restrict*:
assumes $\text{refl-on}\ A\ r$
shows $\text{refl-on}\ (A \cap B)\ (r \cap B \times B)$
 $\langle \text{proof} \rangle$

lemma *sym-allrel*: $\text{sym}\ (A \times A)$
 $\langle \text{proof} \rangle$

lemma *sym-restrict*:
assumes $\text{sym}\ r$
shows $\text{sym}\ (r \cap A \times A)$
 $\langle \text{proof} \rangle$

lemma *trans-allrel*: $\text{trans}\ (A \times A)$
 $\langle \text{proof} \rangle$

lemma *equiv-Int*:
assumes $\text{equiv}\ A\ r$ **and** $\text{equiv}\ B\ s$
shows $\text{equiv}\ (A \cap B)\ (r \cap s)$
 $\langle \text{proof} \rangle$

lemma *equiv-allrel*: $\text{equiv}\ A\ (A \times A)$
 $\langle \text{proof} \rangle$

lemma *equiv-restrict*:
assumes *equiv A r*
shows *equiv (A ∩ B) (r ∩ B × B)*
<proof>

lemma *invertible-times-eq-zero*:
fixes $x :: \text{real}^n$ **and** $A :: \text{real}^n$
assumes *invertible A and A *v x = 0*
shows $x = 0$
<proof>

lemma *times-invertible-eq-zero*:
fixes $x :: \text{real}^n$ **and** $A :: \text{real}^n$
assumes *invertible A and x v* A = 0*
shows $x = 0$
<proof>

lemma *matrix-id-invertible*:
invertible (mat 1 :: ('a::semiring-1)^n)
<proof>

lemma *Image-refl-on-nonempty*:
assumes *refl-on A r and x ∈ A*
shows $x \in r^{\{x\}}$
<proof>

lemma *quotient-element-nonempty*:
assumes *equiv A r and X ∈ A//r*
shows $\exists x. x \in X$
<proof>

lemma *zero-3*: $(3::3) = 0$
<proof>

lemma *card-suc-ge-insert*:
fixes A **and** x
shows $\text{card } A + 1 \geq \text{card } (\text{insert } x A)$
<proof>

lemma *card-le-UNIV*:
fixes $A :: ('n::\text{finite}) \text{ set}$
shows $\text{card } A \leq \text{CARD}('n)$
<proof>

lemma *partition-Image-element*:
assumes *equiv A r and X ∈ A//r and x ∈ X*
shows $r^{\{x\}} = X$
<proof>

lemma *card-insert-ge*: $\text{card} (\text{insert } x \ A) \geq \text{card } A$
<proof>

lemma *choose-1*:
 assumes $\text{card } S = 1$
 shows $\exists x. S = \{x\}$
<proof>

lemma *choose-2*:
 assumes $\text{card } S = 2$
 shows $\exists x \ y. S = \{x,y\}$
<proof>

lemma *choose-3*:
 assumes $\text{card } S = 3$
 shows $\exists x \ y \ z. S = \{x,y,z\}$
<proof>

lemma *card-gt-0-diff-singleton*:
 assumes $\text{card } S > 0$ **and** $x \in S$
 shows $\text{card} (S - \{x\}) = \text{card } S - 1$
<proof>

lemma *eq-3-or-of-3*:
 fixes $j :: 4$
 shows $j = 3 \vee (\exists j'::3. j = \text{of-int} (\text{Rep-bit1 } j'))$
<proof>

lemma *sgn-plus*:
 fixes $x \ y :: 'a::\text{linordered-idom}$
 assumes $\text{sgn } x = \text{sgn } y$
 shows $\text{sgn} (x + y) = \text{sgn } x$
<proof>

lemma *sgn-div*:
 fixes $x \ y :: 'a::\text{linordered-field}$
 assumes $y \neq 0$ **and** $\text{sgn } x = \text{sgn } y$
 shows $x / y > 0$
<proof>

lemma *abs-plus*:
 fixes $x \ y :: 'a::\text{linordered-idom}$
 assumes $\text{sgn } x = \text{sgn } y$
 shows $|x + y| = |x| + |y|$
<proof>

lemma *sgn-plus-abs*:
 fixes $x \ y :: 'a::\text{linordered-idom}$

assumes $|x| > |y|$
shows $\text{sgn}(x + y) = \text{sgn } x$
 {proof}

end

3 Tarski's geometry

theory *Tarski*
imports *Complex-Main Miscellany Metric*
begin

3.1 The axioms

The axioms, and all theorems beginning with *th* followed by a number, are based on corresponding axioms and theorems in [3].

locale *tarski-first3* =
fixes $C :: 'p \Rightarrow 'p \Rightarrow 'p \Rightarrow 'p \Rightarrow \text{bool}$ ($\langle \cdot - \equiv - \cdot \rangle$ [99,99,99,99] 50)
assumes $A1: \forall a b. a b \equiv b a$
and $A2: \forall a b p q r s. a b \equiv p q \wedge a b \equiv r s \longrightarrow p q \equiv r s$
and $A3: \forall a b c. a b \equiv c c \longrightarrow a = b$

locale *tarski-first5* = *tarski-first3* +
fixes $B :: 'p \Rightarrow 'p \Rightarrow 'p \Rightarrow \text{bool}$
assumes $A4: \forall q a b c. \exists x. B q a x \wedge a x \equiv b c$
and $A5: \forall a b c d a' b' c' d'. a \neq b \wedge B a b c \wedge B a' b' c'$
 $\wedge a b \equiv a' b' \wedge b c \equiv b' c' \wedge a d \equiv a' d' \wedge b$
 $d \equiv b' d'$
 $\longrightarrow c d \equiv c' d'$

locale *tarski-absolute-space* = *tarski-first5* +
assumes $A6: \forall a b. B a b a \longrightarrow a = b$
and $A7: \forall a b c p q. B a p c \wedge B b q c \longrightarrow (\exists x. B p x b \wedge B q x a)$
and $A11: \forall X Y. (\exists a. \forall x y. x \in X \wedge y \in Y \longrightarrow B a x y)$
 $\longrightarrow (\exists b. \forall x y. x \in X \wedge y \in Y \longrightarrow B x b y)$

locale *tarski-absolute* = *tarski-absolute-space* +
assumes $A8: \exists a b c. \neg B a b c \wedge \neg B b c a \wedge \neg B c a b$
and $A9: \forall p q a b c. p \neq q \wedge a p \equiv a q \wedge b p \equiv b q \wedge c p \equiv c q$
 $\longrightarrow B a b c \vee B b c a \vee B c a b$

locale *tarski-space* = *tarski-absolute-space* +
assumes $A10: \forall a b c d t. B a d t \wedge B b d c \wedge a \neq d$
 $\longrightarrow (\exists x y. B a b x \wedge B a c y \wedge B x t y)$

locale *tarski* = *tarski-absolute* + *tarski-space*

3.2 Semimetric spaces satisfy the first three axioms

context *semimetric*

begin

definition *smC* :: 'p ⇒ 'p ⇒ 'p ⇒ 'p ⇒ bool (⟨· - ≡_{sm} · -⟩ [99,99,99,99] 50)

where [*simp*]: $a b \equiv_{sm} c d \triangleq dist\ a\ b = dist\ c\ d$

end

sublocale *semimetric* < *tarski-first3 smC*

⟨*proof*⟩

3.3 Some consequences of the first three axioms

context *tarski-first3*

begin

lemma *A1'*: $a b \equiv b a$

⟨*proof*⟩

lemma *A2'*: $\llbracket a b \equiv p\ q; a b \equiv r\ s \rrbracket \implies p\ q \equiv r\ s$

⟨*proof*⟩

lemma *A3'*: $a b \equiv c\ c \implies a = b$

⟨*proof*⟩

theorem *th2-1*: $a b \equiv a b$

⟨*proof*⟩

theorem *th2-2*: $a b \equiv c\ d \implies c\ d \equiv a b$

⟨*proof*⟩

theorem *th2-3*: $\llbracket a b \equiv c\ d; c\ d \equiv e\ f \rrbracket \implies a b \equiv e\ f$

⟨*proof*⟩

theorem *th2-4*: $a b \equiv c\ d \implies b a \equiv c\ d$

⟨*proof*⟩

theorem *th2-5*: $a b \equiv c\ d \implies a b \equiv d\ c$

⟨*proof*⟩

definition *is-segment* :: 'p set ⇒ bool **where**

is-segment $X \triangleq \exists x\ y. X = \{x, y\}$

definition *segments* :: 'p set set **where**

segments = $\{X. is-segment\ X\}$

definition *SC* :: 'p set ⇒ 'p set ⇒ bool **where**

SC $X\ Y \triangleq \exists w\ x\ y\ z. X = \{w, x\} \wedge Y = \{y, z\} \wedge w\ x \equiv y\ z$

definition *SC-rel* :: ('p set × 'p set) set **where**

SC-rel = $\{(X, Y) \mid X\ Y. SC\ X\ Y\}$

lemma *left-segment-congruence*:

assumes $\{a, b\} = \{p, q\}$ **and** $p \equiv c$ $q \equiv d$
shows $a \equiv c$ $b \equiv d$

<proof>

lemma *right-segment-congruence*:

assumes $\{c, d\} = \{p, q\}$ **and** $a \equiv p$ $b \equiv q$
shows $a \equiv c$ $b \equiv d$

<proof>

lemma *C-SC-equiv*: $a \equiv c$ $b \equiv d = SC \{a, b\} \{c, d\}$

<proof>

lemma *SC-rel-subset*: $SC\text{-rel} \subseteq \text{segments} \times \text{segments}$

<proof>

lemmas *SC-refl = th2-1 [simplified]*

lemma *SC-rel-refl*: *refl-on segments SC-rel*

<proof>

lemma *SC-sym*:

assumes $SC \ X \ Y$
shows $SC \ Y \ X$

<proof>

lemma *SC-sym'*: $SC \ X \ Y = SC \ Y \ X$

<proof>

lemma *SC-rel-sym*: *sym SC-rel*

<proof>

lemma *SC-trans*:

assumes $SC \ X \ Y$ **and** $SC \ Y \ Z$
shows $SC \ X \ Z$

<proof>

lemma *SC-rel-trans*: *trans SC-rel*

<proof>

lemma *A3-reversed*:

assumes $a \equiv b$ $b \equiv c$
shows $b = c$

<proof>

lemma *equiv-segments-SC-rel*: *equiv segments SC-rel*

<proof>

end

3.4 Some consequences of the first five axioms

context *tarski-first5*

begin

lemma A_4' : $\exists x. B q a x \wedge a x \equiv b c$
<proof>

theorem *th2-8*: $a a \equiv b b$
<proof>

definition *OFS* :: $['p, 'p, 'p, 'p, 'p, 'p, 'p] \Rightarrow bool$ where
 $OFS a b c d a' b' c' d' \triangleq$
 $B a b c \wedge B a' b' c' \wedge a b \equiv a' b' \wedge b c \equiv b' c' \wedge a d \equiv a' d' \wedge b d \equiv b' d'$

lemma A_5' : $\llbracket OFS a b c d a' b' c' d'; a \neq b \rrbracket \Longrightarrow c d \equiv c' d'$
<proof>

theorem *th2-11*:

assumes *hypotheses*:

$B a b c$

$B a' b' c'$

$a b \equiv a' b'$

$b c \equiv b' c'$

shows $a c \equiv a' c'$
<proof>

lemma *A₄-unique*:

assumes $q \neq a$ and $B q a x$ and $a x \equiv b c$

and $B q a x'$ and $a x' \equiv b c$

shows $x = x'$

<proof>

theorem *th2-12*:

assumes $q \neq a$

shows $\exists! x. B q a x \wedge a x \equiv b c$

<proof>

end

3.5 Simple theorems about betweenness

theorem (in *tarski-first5*) *th3-1*: $B a b b$
<proof>

context *tarski-absolute-space*

begin

lemma A_6' :

assumes $B a b a$

shows $a = b$

<proof>

lemma A7':

assumes $B a p c$ **and** $B b q c$

shows $\exists x. B p x b \wedge B q x a$

<proof>

lemma A11':

assumes $\forall x y. x \in X \wedge y \in Y \longrightarrow B a x y$

shows $\exists b. \forall x y. x \in X \wedge y \in Y \longrightarrow B x b y$

<proof>

theorem th3-2:

assumes $B a b c$

shows $B c b a$

<proof>

theorem th3-4:

assumes $B a b c$ **and** $B b a c$

shows $a = b$

<proof>

theorem th3-5-1:

assumes $B a b d$ **and** $B b c d$

shows $B a b c$

<proof>

theorem th3-6-1:

assumes $B a b c$ **and** $B a c d$

shows $B b c d$

<proof>

theorem th3-7-1:

assumes $b \neq c$ **and** $B a b c$ **and** $B b c d$

shows $B a c d$

<proof>

theorem th3-7-2:

assumes $b \neq c$ **and** $B a b c$ **and** $B b c d$

shows $B a b d$

<proof>

end

3.6 Simple theorems about congruence and betweenness

definition (in *tarski-first5*) $Col :: 'p \Rightarrow 'p \Rightarrow 'p \Rightarrow bool$ where

$Col a b c \triangleq B a b c \vee B b c a \vee B c a b$

end

4 Real Euclidean space and Tarski's axioms

```
theory Euclid-Tarski
imports Tarski
begin
```

4.1 Real Euclidean space satisfies the first five axioms

abbreviation

```
real-euclid-C :: [real^(n::finite), real^(n), real^(n), real^(n)] => bool
( $\langle$  -  $\equiv_{\mathbb{R}}$  -  $\rangle$  [99,99,99,99] 50) where
  real-euclid-C  $\triangleq$  norm-metric.smC
```

definition *real-euclid-B* :: [real^(n::finite), real^(n), real^(n)] => bool

```
( $\langle B_{\mathbb{R}}$  - -  $\rangle$  [99,99,99] 50) where
   $B_{\mathbb{R}}$  a b c  $\triangleq$   $\exists l. 0 \leq l \wedge l \leq 1 \wedge b - a = l *_{\mathbb{R}} (c - a)$ 
```

interpretation *real-euclid*: tarski-first5 *real-euclid-C* *real-euclid-B*
 \langle proof \rangle

4.2 Real Euclidean space also satisfies axioms 6, 7, and 11

lemma *rearrange-real-euclid-B*:

```
  fixes w y z :: real^(n) and h
  shows  $y - w = h *_{\mathbb{R}} (z - w) \longleftrightarrow y = h *_{\mathbb{R}} z + (1 - h) *_{\mathbb{R}} w$ 
 $\langle$ proof $\rangle$ 
```

interpretation *real-euclid*: tarski-absolute-space *real-euclid-C* *real-euclid-B*
 \langle proof \rangle

4.3 Real Euclidean space satisfies the Euclidean axiom

lemma *rearrange-real-euclid-B-2*:

```
  fixes a b c :: real^(n::finite)
  assumes  $l \neq 0$ 
  shows  $b - a = l *_{\mathbb{R}} (c - a) \longleftrightarrow c = (1/l) *_{\mathbb{R}} b + (1 - 1/l) *_{\mathbb{R}} a$ 
 $\langle$ proof $\rangle$ 
```

interpretation *real-euclid*: tarski-space *real-euclid-C* *real-euclid-B*
 \langle proof \rangle

4.4 The real Euclidean plane

lemma *Col-dep2*:

```
  real-euclid.Col a b c  $\longleftrightarrow$  dep2 (b - a) (c - a)
 $\langle$ proof $\rangle$ 
```

lemma *non-Col-example*:

```
   $\neg$ (real-euclid.Col 0 (vector [1/2,0] :: real^2) (vector [0,1/2]))
  (is  $\neg$  (real-euclid.Col ?a ?b ?c))
```

<proof>

interpretation *real-euclid*:

tarski real-euclid-C::([real², real², real², real²] ⇒ bool) real-euclid-B
<proof>

4.5 Special cases of theorems of Tarski's geometry

lemma *real-euclid-B-disjunction*:

assumes $l \geq 0$ **and** $b - a = l *_{\mathbb{R}} (c - a)$

shows $B_{\mathbb{R}} a b c \vee B_{\mathbb{R}} a c b$

<proof>

The following are true in Tarski's geometry, but to prove this would require much more development of it, so only the Euclidean case is proven here.

theorem *real-euclid-th5-1*:

assumes $a \neq b$ **and** $B_{\mathbb{R}} a b c$ **and** $B_{\mathbb{R}} a b d$

shows $B_{\mathbb{R}} a c d \vee B_{\mathbb{R}} a d c$

<proof>

theorem *real-euclid-th5-3*:

assumes $B_{\mathbb{R}} a b d$ **and** $B_{\mathbb{R}} a c d$

shows $B_{\mathbb{R}} a b c \vee B_{\mathbb{R}} a c b$

<proof>

end

5 Linear algebra

theory *Linear-Algebra2*

imports *Miscellany*

begin

lemma *exhaust-4*:

fixes $x :: 4$

shows $x = 1 \vee x = 2 \vee x = 3 \vee x = 4$

<proof>

lemma *forall-4*: $(\forall i::4. P i) \longleftrightarrow P 1 \wedge P 2 \wedge P 3 \wedge P 4$

<proof>

lemma *UNIV-4*: $(UNIV::(4 \text{ set})) = \{1, 2, 3, 4\}$

<proof>

lemma *vector-4*:

fixes $w :: 'a::zero$

shows $(\text{vector } [w, x, y, z] :: 'a^4)\$1 = w$

and $(\text{vector } [w, x, y, z] :: 'a^4)\$2 = x$

and $(\text{vector } [w, x, y, z] :: 'a \text{ } ^4) \$3 = y$
and $(\text{vector } [w, x, y, z] :: 'a \text{ } ^4) \$4 = z$
 $\langle \text{proof} \rangle$

definition

$\text{is-basis} :: (\text{real } ^n) \text{ set} \Rightarrow \text{bool}$ **where**
 $\text{is-basis } S \triangleq \text{independent } S \wedge \text{span } S = \text{UNIV}$

lemma *card-finite*:

assumes $\text{card } S = \text{CARD}('n::\text{finite})$
shows $\text{finite } S$
 $\langle \text{proof} \rangle$

lemma *independent-is-basis*:

fixes $B :: (\text{real } ^n) \text{ set}$
shows $\text{independent } B \wedge \text{card } B = \text{CARD}('n) \longleftrightarrow \text{is-basis } B$
 $\langle \text{proof} \rangle$

lemma *basis-finite*:

fixes $B :: (\text{real } ^n) \text{ set}$
assumes $\text{is-basis } B$
shows $\text{finite } B$
 $\langle \text{proof} \rangle$

lemma *basis-expand*:

assumes $\text{is-basis } B$
shows $\exists c. v = (\sum_{w \in B. (c \ w) *_R \ w})$
 $\langle \text{proof} \rangle$

lemma *not-span-independent-insert*:

fixes $v :: ('a::\text{real-vector}) ^n$
assumes $\text{independent } S$ **and** $v \notin \text{span } S$
shows $\text{independent } (\text{insert } v \ S)$
 $\langle \text{proof} \rangle$

lemma *orthogonal-sum*:

fixes $v :: \text{real } ^n$
assumes $\bigwedge w. w \in S \implies \text{orthogonal } v \ w$
shows $\text{orthogonal } v \ (\sum_{w \in S. c \ w *s \ w})$
 $\langle \text{proof} \rangle$

lemma *orthogonal-self-eq-0*:

fixes $v :: ('a::\text{real-inner}) ^n$
assumes $\text{orthogonal } v \ v$
shows $v = 0$
 $\langle \text{proof} \rangle$

lemma *orthogonal-in-span-eq-0*:

fixes $v :: \text{real } ^n$

assumes $v \in \text{span } S$ **and** $\bigwedge w. w \in S \implies \text{orthogonal } v w$
shows $v = 0$
 $\langle \text{proof} \rangle$

lemma *orthogonal-independent*:

fixes $v :: \text{real}^n$
assumes *independent* S **and** $v \neq 0$ **and** $\bigwedge w. w \in S \implies \text{orthogonal } v w$
shows *independent* $(\text{insert } v S)$
 $\langle \text{proof} \rangle$

lemma *dot-scaleR-mult*:

shows $(k *_R a) \cdot b = k * (a \cdot b)$ **and** $a \cdot (k *_R b) = k * (a \cdot b)$
 $\langle \text{proof} \rangle$

lemma *dependent-explicit-finite*:

fixes $S :: ((\text{'a'}::\{\text{real-vector}, \text{field}\})^n)$ *set*
assumes *finite* S
shows *dependent* $S \longleftrightarrow (\exists u. (\exists v \in S. u v \neq 0) \wedge (\sum v \in S. u v *_R v) = 0)$
 $\langle \text{proof} \rangle$

lemma *dependent-explicit-2*:

fixes $v w :: (\text{'a'}::\{\text{field}, \text{real-vector}\})^n$
assumes $v \neq w$
shows *dependent* $\{v, w\} \longleftrightarrow (\exists i j. (i \neq 0 \vee j \neq 0) \wedge i *_R v + j *_R w = 0)$
 $\langle \text{proof} \rangle$

5.1 Matrices

lemma *zero-not-invertible*:

$\neg (\text{invertible } (0::\text{real}^n)^n)$
 $\langle \text{proof} \rangle$

Based on matrix-vector-column in HOL/Multivariate_Analysis/Euclidean_Space.thy in Isabelle 2009-1:

lemma *vector-matrix-row*:

fixes $x :: (\text{'a'}::\text{comm-semiring-1})^m$ **and** $A :: (\text{'a'}^n)^m$
shows $x v * A = (\sum i \in UNIV. (x \$ i) * s (A \$ i))$
 $\langle \text{proof} \rangle$

lemma *matrix-inv*:

assumes *invertible* M
shows *matrix-inv* $M ** M = \text{mat } 1$
and $M ** \text{matrix-inv } M = \text{mat } 1$
 $\langle \text{proof} \rangle$

lemma *matrix-inv-invertible*:

assumes *invertible* M
shows *invertible* $(\text{matrix-inv } M)$
 $\langle \text{proof} \rangle$

lemma *invertible-times-non-zero*:

fixes $M :: \text{real}^n$
assumes *invertible* M **and** $v \neq 0$
shows $M * v \neq 0$
<proof>

lemma *matrix-right-invertible-ker*:

fixes $M :: \text{real}^{(m::\text{finite})^n}$
shows $(\exists M'. M ** M' = \text{mat } 1) \longleftrightarrow (\forall x. x v * M = 0 \longrightarrow x = 0)$
<proof>

lemma *left-invertible-iff-invertible*:

fixes $M :: \text{real}^n$
shows $(\exists N. N ** M = \text{mat } 1) \longleftrightarrow \text{invertible } M$
<proof>

lemma *right-invertible-iff-invertible*:

fixes $M :: \text{real}^n$
shows $(\exists N. M ** N = \text{mat } 1) \longleftrightarrow \text{invertible } M$
<proof>

definition *symmatrix* :: $'a^n \Rightarrow \text{bool}$ **where**
symmatrix $M \triangleq \text{transpose } M = M$

lemma *symmatrix-preserve*:

fixes $M N :: ('a::\text{comm-semiring-1})^n$
assumes *symmatrix* M
shows *symmatrix* $(N ** M ** \text{transpose } N)$
<proof>

lemma *non-zero-mult-invertible-non-zero*:

fixes $M :: \text{real}^n$
assumes $v \neq 0$ **and** *invertible* M
shows $v v * M \neq 0$
<proof>

end

6 Right group actions

theory *Action*

imports *HOL-Algebra.Group*

begin

locale *action = group +*

fixes $\text{act} :: 'b \Rightarrow 'a \Rightarrow 'b$ (**infixl** $\langle \langle o \rangle \rangle$ 69)
assumes *id-act* [*simp*]: $b \langle o \rangle \mathbf{1} = b$
and *act-act'*:

$g \in \text{carrier } G \wedge h \in \text{carrier } G \longrightarrow (b <_o g) <_o h = b <_o (g \otimes h)$
begin

lemma *act-act*:
assumes $g \in \text{carrier } G$ **and** $h \in \text{carrier } G$
shows $(b <_o g) <_o h = b <_o (g \otimes h)$
 $\langle \text{proof} \rangle$

lemma *act-act-inv [simp]*:
assumes $g \in \text{carrier } G$
shows $b <_o g <_o \text{inv } g = b$
 $\langle \text{proof} \rangle$

lemma *act-inv-act [simp]*:
assumes $g \in \text{carrier } G$
shows $b <_o \text{inv } g <_o g = b$
 $\langle \text{proof} \rangle$

lemma *act-inv-iff*:
assumes $g \in \text{carrier } G$
shows $b <_o \text{inv } g = c \longleftrightarrow b = c <_o g$
 $\langle \text{proof} \rangle$

end

end

7 Projective geometry

theory *Projective*
imports *Linear-Algebra2*
Euclid-Tarski
Action
begin

7.1 Proportionality on non-zero vectors

context *vector-space*
begin

definition *proportionality* :: $({}^b \times {}^b)$ set **where**
 $\text{proportionality} \triangleq \{(x, y). x \neq 0 \wedge y \neq 0 \wedge (\exists k. x = \text{scale } k \ y)\}$

definition *non-zero-vectors* :: b set **where**
 $\text{non-zero-vectors} \triangleq \{x. x \neq 0\}$

lemma *proportionality-subset*:
 $\text{local.proportionality} \subseteq \text{local.non-zero-vectors} \times \text{local.non-zero-vectors}$
 $\langle \text{proof} \rangle$

lemma *proportionality-refl-on*: *refl-on local.non-zero-vectors local.proportionality*
⟨*proof*⟩

lemma *proportionality-sym*: *sym local.proportionality*
⟨*proof*⟩

lemma *proportionality-trans*: *trans local.proportionality*
⟨*proof*⟩

theorem *proportionality-equiv*: *equiv local.non-zero-vectors local.proportionality*
⟨*proof*⟩

end

definition *invertible-proportionality* ::
((*real*^(*n::finite*)^{*n*}) × (*real*^(*n*)^{*n*})) **set where**
invertible-proportionality ≜
real-vector.proportionality ∩ (*Collect invertible* × *Collect invertible*)

lemma *invertible-proportionality-equiv*:
equiv (Collect invertible :: (real^(*n::finite*)^{*n*}) *set)*
invertible-proportionality
(**is** *equiv ?invs -*)
⟨*proof*⟩

7.2 Points of the real projective plane

typedef *proj2* = (*real-vector.non-zero-vectors :: (real*^3) *set*) // *real-vector.proportionality*
⟨*proof*⟩

definition *proj2-rep* :: *proj2* ⇒ *real*^3 **where**
proj2-rep *x* ≜ *ε v. v ∈ Rep-proj2 x*

definition *proj2-abs* :: *real*^3 ⇒ *proj2* **where**
proj2-abs *v* ≜ *Abs-proj2 (real-vector.proportionality “ {v})*

lemma *proj2-rep-in*: *proj2-rep* *x* ∈ *Rep-proj2* *x*
⟨*proof*⟩

lemma *proj2-rep-non-zero*: *proj2-rep* *x* ≠ 0
⟨*proof*⟩

lemma *proj2-rep-abs*:
fixes *v :: real*^3
assumes *v* ∈ *real-vector.non-zero-vectors*
shows (*v*, *proj2-rep (proj2-abs v)*) ∈ *real-vector.proportionality*
⟨*proof*⟩

lemma *proj2-abs-rep*: $\text{proj2-abs } (\text{proj2-rep } x) = x$
 ⟨*proof*⟩

lemma *proj2-abs-mult*:
 assumes $c \neq 0$
 shows $\text{proj2-abs } (c *_{\mathbb{R}} v) = \text{proj2-abs } v$
 ⟨*proof*⟩

lemma *proj2-abs-mult-rep*:
 assumes $c \neq 0$
 shows $\text{proj2-abs } (c *_{\mathbb{R}} \text{proj2-rep } x) = x$
 ⟨*proof*⟩

lemma *proj2-rep-inj*: $\text{inj } \text{proj2-rep}$
 ⟨*proof*⟩

lemma *proj2-rep-abs2*:
 assumes $v \neq 0$
 shows $\exists k. k \neq 0 \wedge \text{proj2-rep } (\text{proj2-abs } v) = k *_{\mathbb{R}} v$
 ⟨*proof*⟩

lemma *proj2-abs-abs-mult*:
 assumes $\text{proj2-abs } v = \text{proj2-abs } w$ and $w \neq 0$
 shows $\exists c. v = c *_{\mathbb{R}} w$
 ⟨*proof*⟩

lemma *dependent-proj2-abs*:
 assumes $p \neq 0$ and $q \neq 0$ and $i \neq 0 \vee j \neq 0$ and $i *_{\mathbb{R}} p + j *_{\mathbb{R}} q = 0$
 shows $\text{proj2-abs } p = \text{proj2-abs } q$
 ⟨*proof*⟩

lemma *proj2-rep-dependent*:
 assumes $i *_{\mathbb{R}} \text{proj2-rep } v + j *_{\mathbb{R}} \text{proj2-rep } w = 0$
 (is $i *_{\mathbb{R}} ?p + j *_{\mathbb{R}} ?q = 0$)
 and $i \neq 0 \vee j \neq 0$
 shows $v = w$
 ⟨*proof*⟩

lemma *proj2-rep-independent*:
 assumes $p \neq q$
 shows $\text{independent } \{\text{proj2-rep } p, \text{proj2-rep } q\}$
 ⟨*proof*⟩

7.3 Lines of the real projective plane

definition *proj2-Col* :: $[\text{proj2}, \text{proj2}, \text{proj2}] \Rightarrow \text{bool}$ where
 $\text{proj2-Col } p \ q \ r \triangleq$
 $(\exists i \ j \ k. i *_{\mathbb{R}} \text{proj2-rep } p + j *_{\mathbb{R}} \text{proj2-rep } q + k *_{\mathbb{R}} \text{proj2-rep } r = 0$
 $\wedge (i \neq 0 \vee j \neq 0 \vee k \neq 0))$

lemma *proj2-Col-abs*:

assumes $p \neq 0$ **and** $q \neq 0$ **and** $r \neq 0$ **and** $i \neq 0 \vee j \neq 0 \vee k \neq 0$
and $i *_R p + j *_R q + k *_R r = 0$
shows *proj2-Col* (*proj2-abs* p) (*proj2-abs* q) (*proj2-abs* r)
(**is** *proj2-Col* $?pp$ $?pq$ $?pr$)

<proof>

lemma *proj2-Col-permute*:

assumes *proj2-Col* a b c
shows *proj2-Col* a c b
and *proj2-Col* b a c

<proof>

lemma *proj2-Col-coincide*: *proj2-Col* a a c

<proof>

lemma *proj2-Col-iff*:

assumes $a \neq r$
shows *proj2-Col* a r $t \iff$
 $t = a \vee (\exists i. t = \text{proj2-abs } (i *_R (\text{proj2-rep } a) + (\text{proj2-rep } r)))$

<proof>

definition *proj2-Col-coeff* :: *proj2* \Rightarrow *proj2* \Rightarrow *proj2* \Rightarrow *real* **where**

proj2-Col-coeff a r $t \triangleq \epsilon i. t = \text{proj2-abs } (i *_R \text{proj2-rep } a + \text{proj2-rep } r)$

lemma *proj2-Col-coeff*:

assumes *proj2-Col* a r t **and** $a \neq r$ **and** $t \neq a$
shows $t = \text{proj2-abs } ((\text{proj2-Col-coeff } a$ r $t) *_R \text{proj2-rep } a + \text{proj2-rep } r)$

<proof>

lemma *proj2-Col-coeff-unique'*:

assumes $a \neq 0$ **and** $r \neq 0$ **and** $\text{proj2-abs } a \neq \text{proj2-abs } r$
and $\text{proj2-abs } (i *_R a + r) = \text{proj2-abs } (j *_R a + r)$
shows $i = j$

<proof>

lemma *proj2-Col-coeff-unique*:

assumes $a \neq r$
and $\text{proj2-abs } (i *_R \text{proj2-rep } a + \text{proj2-rep } r)$
 $= \text{proj2-abs } (j *_R \text{proj2-rep } a + \text{proj2-rep } r)$
shows $i = j$

<proof>

datatype *proj2-line* = *P2L* *proj2*

definition *L2P* :: *proj2-line* \Rightarrow *proj2* **where**

L2P $l \triangleq \text{case } l \text{ of } \text{P2L } p \Rightarrow p$

lemma *L2P-P2L* [*simp*]: $L2P (P2L p) = p$
 ⟨*proof*⟩

lemma *P2L-L2P* [*simp*]: $P2L (L2P l) = l$
 ⟨*proof*⟩

lemma *L2P-inj* [*simp*]:
assumes $L2P l = L2P m$
shows $l = m$
 ⟨*proof*⟩

lemma *P2L-to-L2P*: $P2L p = l \longleftrightarrow p = L2P l$
 ⟨*proof*⟩

definition *proj2-line-abs* :: $real^3 \Rightarrow proj2-line$ **where**
proj2-line-abs $v \triangleq P2L (proj2-abs v)$

definition *proj2-line-rep* :: $proj2-line \Rightarrow real^3$ **where**
proj2-line-rep $l \triangleq proj2-rep (L2P l)$

lemma *proj2-line-rep-abs*:
assumes $v \neq 0$
shows $\exists k. k \neq 0 \wedge proj2-line-rep (proj2-line-abs v) = k *_R v$
 ⟨*proof*⟩

lemma *proj2-line-abs-rep* [*simp*]: $proj2-line-abs (proj2-line-rep l) = l$
 ⟨*proof*⟩

lemma *proj2-line-rep-non-zero*: $proj2-line-rep l \neq 0$
 ⟨*proof*⟩

lemma *proj2-line-rep-dependent*:
assumes $i *_R proj2-line-rep l + j *_R proj2-line-rep m = 0$
and $i \neq 0 \vee j \neq 0$
shows $l = m$
 ⟨*proof*⟩

lemma *proj2-line-abs-mult*:
assumes $k \neq 0$
shows $proj2-line-abs (k *_R v) = proj2-line-abs v$
 ⟨*proof*⟩

lemma *proj2-line-abs-abs-mult*:
assumes $proj2-line-abs v = proj2-line-abs w$ **and** $w \neq 0$
shows $\exists k. v = k *_R w$
 ⟨*proof*⟩

definition *proj2-incident* :: $proj2 \Rightarrow proj2-line \Rightarrow bool$ **where**
proj2-incident $p l \triangleq (proj2-rep p) \cdot (proj2-line-rep l) = 0$

lemma *proj2-points-define-line*:

shows $\exists l. \text{proj2-incident } p \ l \wedge \text{proj2-incident } q \ l$
(*proof*)

definition *proj2-line-through* :: *proj2* \Rightarrow *proj2* \Rightarrow *proj2-line* **where**
proj2-line-through $p \ q \triangleq \epsilon \ l. \text{proj2-incident } p \ l \wedge \text{proj2-incident } q \ l$

lemma *proj2-line-through-incident*:

shows *proj2-incident* $p \ (\text{proj2-line-through } p \ q)$
and *proj2-incident* $q \ (\text{proj2-line-through } p \ q)$
(*proof*)

lemma *proj2-line-through-unique*:

assumes $p \neq q$ **and** *proj2-incident* $p \ l$ **and** *proj2-incident* $q \ l$
shows $l = \text{proj2-line-through } p \ q$
(*proof*)

lemma *proj2-incident-unique*:

assumes *proj2-incident* $p \ l$
and *proj2-incident* $q \ l$
and *proj2-incident* $p \ m$
and *proj2-incident* $q \ m$
shows $p = q \vee l = m$
(*proof*)

lemma *proj2-lines-define-point*: $\exists p. \text{proj2-incident } p \ l \wedge \text{proj2-incident } p \ m$
(*proof*)

definition *proj2-intersection* :: *proj2-line* \Rightarrow *proj2-line* \Rightarrow *proj2* **where**
proj2-intersection $l \ m \triangleq L2P \ (\text{proj2-line-through } (L2P \ l) \ (L2P \ m))$

lemma *proj2-incident-switch*:

assumes *proj2-incident* $p \ l$
shows *proj2-incident* $(L2P \ l) \ (P2L \ p)$
(*proof*)

lemma *proj2-intersection-incident*:

shows *proj2-incident* $(\text{proj2-intersection } l \ m) \ l$
and *proj2-incident* $(\text{proj2-intersection } l \ m) \ m$
(*proof*)

lemma *proj2-intersection-unique*:

assumes $l \neq m$ **and** *proj2-incident* $p \ l$ **and** *proj2-incident* $p \ m$
shows $p = \text{proj2-intersection } l \ m$
(*proof*)

lemma *proj2-not-self-incident*:

$\neg (\text{proj2-incident } p \ (P2L \ p))$

$\langle \text{proof} \rangle$

lemma *proj2-another-point-on-line*:

$\exists q. q \neq p \wedge \text{proj2-incident } q \ l$

$\langle \text{proof} \rangle$

lemma *proj2-another-line-through-point*:

$\exists m. m \neq l \wedge \text{proj2-incident } p \ m$

$\langle \text{proof} \rangle$

lemma *proj2-incident-abs*:

assumes $v \neq 0$ **and** $w \neq 0$

shows $\text{proj2-incident } (\text{proj2-abs } v) (\text{proj2-line-abs } w) \longleftrightarrow v \cdot w = 0$

$\langle \text{proof} \rangle$

lemma *proj2-incident-left-abs*:

assumes $v \neq 0$

shows $\text{proj2-incident } (\text{proj2-abs } v) \ l \longleftrightarrow v \cdot (\text{proj2-line-rep } l) = 0$

$\langle \text{proof} \rangle$

lemma *proj2-incident-right-abs*:

assumes $v \neq 0$

shows $\text{proj2-incident } p (\text{proj2-line-abs } v) \longleftrightarrow (\text{proj2-rep } p) \cdot v = 0$

$\langle \text{proof} \rangle$

definition *proj2-set-Col* :: $\text{proj2 set} \Rightarrow \text{bool}$ **where**

$\text{proj2-set-Col } S \triangleq \exists l. \forall p \in S. \text{proj2-incident } p \ l$

lemma *proj2-subset-Col*:

assumes $T \subseteq S$ **and** $\text{proj2-set-Col } S$

shows $\text{proj2-set-Col } T$

$\langle \text{proof} \rangle$

definition *proj2-no-3-Col* :: $\text{proj2 set} \Rightarrow \text{bool}$ **where**

$\text{proj2-no-3-Col } S \triangleq \text{card } S = 4 \wedge (\forall p \in S. \neg \text{proj2-set-Col } (S - \{p\}))$

lemma *proj2-Col-iff-not-invertible*:

$\text{proj2-Col } p \ q \ r$

$\longleftrightarrow \neg \text{invertible } (\text{vector } [\text{proj2-rep } p, \text{proj2-rep } q, \text{proj2-rep } r] :: \text{real}^{\wedge 3})$

(**is** $\longleftrightarrow \neg \text{invertible } (\text{vector } [?u, ?v, ?w])$)

$\langle \text{proof} \rangle$

lemma *not-invertible-iff-proj2-set-Col*:

$\neg \text{invertible } (\text{vector } [\text{proj2-rep } p, \text{proj2-rep } q, \text{proj2-rep } r] :: \text{real}^{\wedge 3})$

$\longleftrightarrow \text{proj2-set-Col } \{p, q, r\}$

(**is** $\neg \text{invertible } ?M \longleftrightarrow -$)

$\langle \text{proof} \rangle$

lemma *proj2-Col-iff-set-Col*:

proj2-Col $p \ q \ r \iff \text{proj2-set-Col } \{p,q,r\}$
 ⟨proof⟩

lemma *proj2-incident-Col*:
 assumes *proj2-incident* $p \ l$ and *proj2-incident* $q \ l$ and *proj2-incident* $r \ l$
 shows *proj2-Col* $p \ q \ r$
 ⟨proof⟩

lemma *proj2-incident-iff-Col*:
 assumes $p \neq q$ and *proj2-incident* $p \ l$ and *proj2-incident* $q \ l$
 shows *proj2-incident* $r \ l \iff \text{proj2-Col } p \ q \ r$
 ⟨proof⟩

lemma *proj2-incident-iff*:
 assumes $p \neq q$ and *proj2-incident* $p \ l$ and *proj2-incident* $q \ l$
 shows *proj2-incident* $r \ l$
 $\iff r = p \vee (\exists k. r = \text{proj2-abs } (k *_R \text{proj2-rep } p + \text{proj2-rep } q))$
 ⟨proof⟩

lemma *not-proj2-set-Col-iff-span*:
 assumes $\text{card } S = 3$
 shows $\neg \text{proj2-set-Col } S \iff \text{span } (\text{proj2-rep } ` S) = \text{UNIV}$
 ⟨proof⟩

lemma *proj2-no-3-Col-span*:
 assumes *proj2-no-3-Col* S and $p \in S$
 shows $\text{span } (\text{proj2-rep } `(S - \{p\})) = \text{UNIV}$
 ⟨proof⟩

lemma *fourth-proj2-no-3-Col*:
 assumes $\neg \text{proj2-Col } p \ q \ r$
 shows $\exists s. \text{proj2-no-3-Col } \{s,r,p,q\}$
 ⟨proof⟩

lemma *proj2-set-Col-expand*:
 assumes *proj2-set-Col* S and $\{p,q,r\} \subseteq S$ and $p \neq q$ and $r \neq p$
 shows $\exists k. r = \text{proj2-abs } (k *_R \text{proj2-rep } p + \text{proj2-rep } q)$
 ⟨proof⟩

7.4 Collineations of the real projective plane

typedef *cltn2* =
 (Collect invertible :: (real³) set)//invertible-proportionality
 ⟨proof⟩

definition *cltn2-rep* :: *cltn2* \Rightarrow real³ where
cltn2-rep $A \triangleq \epsilon \ B. B \in \text{Rep-cltn2 } A$

definition *cltn2-abs* :: real³ \Rightarrow *cltn2* where

$cltn2-abs\ B \triangleq Abs-cltn2\ (invertible-proportionality\ \{\{B\}\})$

definition $cltn2-independent :: cltn2\ set \Rightarrow bool$ **where**
 $cltn2-independent\ X \triangleq independent\ \{cltn2-rep\ A \mid A.\ A \in X\}$

definition $apply-cltn2 :: proj2 \Rightarrow cltn2 \Rightarrow proj2$ **where**
 $apply-cltn2\ x\ A \triangleq proj2-abs\ (proj2-rep\ x\ v*cltn2-rep\ A)$

lemma $cltn2-rep-in: cltn2-rep\ B \in Rep-cltn2\ B$
 $\langle proof \rangle$

lemma $cltn2-rep-invertible: invertible\ (cltn2-rep\ A)$
 $\langle proof \rangle$

lemma $cltn2-rep-abs:$
fixes $A :: real^3^3$
assumes $invertible\ A$
shows $(A, cltn2-rep\ (cltn2-abs\ A)) \in invertible-proportionality$
 $\langle proof \rangle$

lemma $cltn2-rep-abs2:$
assumes $invertible\ A$
shows $\exists k. k \neq 0 \wedge cltn2-rep\ (cltn2-abs\ A) = k *_R A$
 $\langle proof \rangle$

lemma $cltn2-abs-rep: cltn2-abs\ (cltn2-rep\ A) = A$
 $\langle proof \rangle$

lemma $cltn2-abs-mult:$
assumes $k \neq 0$ **and** $invertible\ A$
shows $cltn2-abs\ (k *_R A) = cltn2-abs\ A$
 $\langle proof \rangle$

lemma $cltn2-abs-mult-rep:$
assumes $k \neq 0$
shows $cltn2-abs\ (k *_R cltn2-rep\ A) = A$
 $\langle proof \rangle$

lemma $apply-cltn2-abs:$
assumes $x \neq 0$ **and** $invertible\ A$
shows $apply-cltn2\ (proj2-abs\ x)\ (cltn2-abs\ A) = proj2-abs\ (x\ v* A)$
 $\langle proof \rangle$

lemma $apply-cltn2-left-abs:$
assumes $v \neq 0$
shows $apply-cltn2\ (proj2-abs\ v)\ C = proj2-abs\ (v\ v* cltn2-rep\ C)$
 $\langle proof \rangle$

lemma $apply-cltn2-right-abs:$

assumes *invertible M*
shows $\text{apply-cltn2 } p \text{ (cltn2-abs } M) = \text{proj2-abs (proj2-rep } p \text{ } v * M)$
 $\langle \text{proof} \rangle$

lemma *non-zero-mult-rep-non-zero:*

assumes $v \neq 0$
shows $v * \text{cltn2-rep } C \neq 0$
 $\langle \text{proof} \rangle$

lemma *rep-mult-rep-non-zero: proj2-rep p v* cltn2-rep A ≠ 0*

$\langle \text{proof} \rangle$

definition *cltn2-image* :: *proj2 set* \Rightarrow *cltn2* \Rightarrow *proj2 set* **where**
 $\text{cltn2-image } P \ A \triangleq \{ \text{apply-cltn2 } p \ A \mid p. p \in P \}$

7.4.1 As a group

definition *cltn2-id* :: *cltn2* **where**

$\text{cltn2-id} \triangleq \text{cltn2-abs (mat 1)}$

definition *cltn2-compose* :: *cltn2* \Rightarrow *cltn2* \Rightarrow *cltn2* **where**

$\text{cltn2-compose } A \ B \triangleq \text{cltn2-abs (cltn2-rep } A \ ** \ \text{cltn2-rep } B)$

definition *cltn2-inverse* :: *cltn2* \Rightarrow *cltn2* **where**

$\text{cltn2-inverse } A \triangleq \text{cltn2-abs (matrix-inv (cltn2-rep } A))$

lemma *cltn2-compose-abs:*

assumes *invertible M and invertible N*
shows $\text{cltn2-compose (cltn2-abs } M) \text{ (cltn2-abs } N) = \text{cltn2-abs (} M \ ** \ N)$
 $\langle \text{proof} \rangle$

lemma *cltn2-compose-left-abs:*

assumes *invertible M*
shows $\text{cltn2-compose (cltn2-abs } M) \ A = \text{cltn2-abs (} M \ ** \ \text{cltn2-rep } A)$
 $\langle \text{proof} \rangle$

lemma *cltn2-compose-right-abs:*

assumes *invertible M*
shows $\text{cltn2-compose } A \text{ (cltn2-abs } M) = \text{cltn2-abs (cltn2-rep } A \ ** \ M)$
 $\langle \text{proof} \rangle$

lemma *cltn2-abs-rep-abs-mult:*

assumes *invertible M and invertible N*
shows $\text{cltn2-abs (cltn2-rep (cltn2-abs } M) \ ** \ N) = \text{cltn2-abs (} M \ ** \ N)$
 $\langle \text{proof} \rangle$

lemma *cltn2-assoc:*

$\text{cltn2-compose (cltn2-compose } A \ B) \ C = \text{cltn2-compose } A \ \text{(cltn2-compose } B \ C)$
 $\langle \text{proof} \rangle$

lemma *cltn2-left-id*: *cltn2-compose cltn2-id A = A*
<proof>

lemma *cltn2-left-inverse*: *cltn2-compose (cltn2-inverse A) A = cltn2-id*
<proof>

lemma *cltn2-left-inverse-ex*:
 $\exists B. \text{cltn2-compose } B \ A = \text{cltn2-id}$
<proof>

interpretation *cltn2*:
group ($| \text{carrier} = \text{UNIV}, \text{mult} = \text{cltn2-compose}, \text{one} = \text{cltn2-id} |$)
<proof>

lemma *cltn2-inverse-inv* [*simp*]:
 $\text{inv}(| \text{carrier} = \text{UNIV}, \text{mult} = \text{cltn2-compose}, \text{one} = \text{cltn2-id} |) \ A$
 $= \text{cltn2-inverse } A$
<proof>

lemmas *cltn2-inverse-id* [*simp*] = *cltn2.inv-one* [*simplified*]
and *cltn2-inverse-compose* = *cltn2.inv-mult-group* [*simplified*]

7.4.2 As a group action

lemma *apply-cltn2-id* [*simp*]: *apply-cltn2 p cltn2-id = p*
<proof>

lemma *apply-cltn2-compose*:
 $\text{apply-cltn2} (\text{apply-cltn2 } p \ A) \ B = \text{apply-cltn2 } p (\text{cltn2-compose } A \ B)$
<proof>

interpretation *cltn2*:
action ($| \text{carrier} = \text{UNIV}, \text{mult} = \text{cltn2-compose}, \text{one} = \text{cltn2-id} |$) *apply-cltn2*
<proof>

definition *cltn2-transpose* :: *cltn2* \Rightarrow *cltn2* **where**
 $\text{cltn2-transpose } A \triangleq \text{cltn2-abs} (\text{transpose} (\text{cltn2-rep } A))$

definition *apply-cltn2-line* :: *proj2-line* \Rightarrow *cltn2* \Rightarrow *proj2-line* **where**
 $\text{apply-cltn2-line } l \ A$
 $\triangleq P2L (\text{apply-cltn2} (L2P \ l) (\text{cltn2-transpose} (\text{cltn2-inverse } A)))$

lemma *cltn2-transpose-abs*:
assumes *invertible M*
shows $\text{cltn2-transpose} (\text{cltn2-abs } M) = \text{cltn2-abs} (\text{transpose } M)$
<proof>

lemma *cltn2-transpose-compose*:

$cltn2-transpose (cltn2-compose A B)$
 $= cltn2-compose (cltn2-transpose B) (cltn2-transpose A)$
 <proof>

lemma *cltn2-transpose-transpose*: $cltn2-transpose (cltn2-transpose A) = A$
 <proof>

lemma *cltn2-transpose-id [simp]*: $cltn2-transpose cltn2-id = cltn2-id$
 <proof>

lemma *apply-cltn2-line-id [simp]*: $apply-cltn2-line l cltn2-id = l$
 <proof>

lemma *apply-cltn2-line-compose*:
 $apply-cltn2-line (apply-cltn2-line l A) B$
 $= apply-cltn2-line l (cltn2-compose A B)$
 <proof>

interpretation *cltn2-line*:
action
 (|carrier = UNIV, mult = cltn2-compose, one = cltn2-id|)
apply-cltn2-line
 <proof>

lemmas *apply-cltn2-inv [simp]* = *cltn2.act-act-inv [simplified]*
lemmas *apply-cltn2-line-inv [simp]* = *cltn2-line.act-act-inv [simplified]*

lemma *apply-cltn2-line-alt-def*:
 $apply-cltn2-line l A$
 $= proj2-line-abs (cltn2-rep (cltn2-inverse A) *v proj2-line-rep l)$
 <proof>

lemma *rep-mult-line-rep-non-zero*: $cltn2-rep A *v proj2-line-rep l \neq 0$
 <proof>

lemma *apply-cltn2-incident*:
 $proj2-incident p (apply-cltn2-line l A)$
 $\longleftrightarrow proj2-incident (apply-cltn2 p (cltn2-inverse A)) l$
 <proof>

lemma *apply-cltn2-preserve-incident [iff]*:
 $proj2-incident (apply-cltn2 p A) (apply-cltn2-line l A)$
 $\longleftrightarrow proj2-incident p l$
 <proof>

lemma *apply-cltn2-preserve-set-Col*:
assumes *proj2-set-Col S*
shows *proj2-set-Col {apply-cltn2 p C | p. p ∈ S}*
 <proof>

lemma *apply-cltn2-injective*:
assumes *apply-cltn2 p C = apply-cltn2 q C*
shows $p = q$
 \langle *proof* \rangle

lemma *apply-cltn2-line-injective*:
assumes *apply-cltn2-line l C = apply-cltn2-line m C*
shows $l = m$
 \langle *proof* \rangle

lemma *apply-cltn2-line-unique*:
assumes $p \neq q$ **and** *proj2-incident p l* **and** *proj2-incident q l*
and *proj2-incident (apply-cltn2 p C) m*
and *proj2-incident (apply-cltn2 q C) m*
shows *apply-cltn2-line l C = m*
 \langle *proof* \rangle

lemma *apply-cltn2-unique*:
assumes $l \neq m$ **and** *proj2-incident p l* **and** *proj2-incident p m*
and *proj2-incident q (apply-cltn2-line l C)*
and *proj2-incident q (apply-cltn2-line m C)*
shows *apply-cltn2 p C = q*
 \langle *proof* \rangle

7.4.3 Parts of some Statements from [1]

All theorems with names beginning with *statement* are based on corresponding theorems in [1].

lemma *statement52-existence*:
fixes $a :: \text{proj2}^{\wedge 3}$ **and** $a3 :: \text{proj2}$
assumes *proj2-no-3-Col (insert a3 (range (($\$$) a)))*
shows $\exists A. \text{apply-cltn2} (\text{proj2-abs} (\text{vector } [1,1,1])) A = a3 \wedge$
 $(\forall j. \text{apply-cltn2} (\text{proj2-abs} (\text{axis } j 1)) A = a\$j)$
 \langle *proof* \rangle

lemma *statement53-existence*:
fixes $p :: \text{proj2}^{\wedge 4 \wedge 2}$
assumes $\forall i. \text{proj2-no-3-Col} (\text{range} ((\text{\$}) (p\$i)))$
shows $\exists C. \forall j. \text{apply-cltn2} (p\$0\$j) C = p\$1\$j$
 \langle *proof* \rangle

lemma *apply-cltn2-linear*:
assumes $j *_R v + k *_R w \neq 0$
shows $j *_R (v \text{ v* } \text{cltn2-rep } C) + k *_R (w \text{ v* } \text{cltn2-rep } C) \neq 0$
(is ?u $\neq 0$)
and *apply-cltn2 (proj2-abs (j *_R v + k *_R w)) C*
 $= \text{proj2-abs} (j *_R (v \text{ v* } \text{cltn2-rep } C) + k *_R (w \text{ v* } \text{cltn2-rep } C))$
 \langle *proof* \rangle

lemma *apply-cltn2-imp-mult*:
assumes *apply-cltn2* p $C = q$
shows $\exists k. k \neq 0 \wedge \text{proj2-rep } p \ v * \text{cltn2-rep } C = k *_R \text{proj2-rep } q$
 $\langle \text{proof} \rangle$

lemma *statement55*:
assumes $p \neq q$
and *apply-cltn2* p $C = q$
and *apply-cltn2* q $C = p$
and *proj2-incident* p l
and *proj2-incident* q l
and *proj2-incident* r l
shows *apply-cltn2* (*apply-cltn2* r C) $C = r$
 $\langle \text{proof} \rangle$

7.5 Cross ratios

definition *cross-ratio* :: *proj2* \Rightarrow *proj2* \Rightarrow *proj2* \Rightarrow *proj2* \Rightarrow *real* **where**
cross-ratio p q r $s \triangleq \text{proj2-Col-coeff } p$ q s / $\text{proj2-Col-coeff } p$ q r

definition *cross-ratio-correct* :: *proj2* \Rightarrow *proj2* \Rightarrow *proj2* \Rightarrow *proj2* \Rightarrow *bool* **where**
cross-ratio-correct p q r $s \triangleq$
proj2-set-Col $\{p, q, r, s\} \wedge p \neq q \wedge r \neq p \wedge s \neq p \wedge r \neq q$

lemma *proj2-Col-coeff-abs*:
assumes $p \neq q$ **and** $j \neq 0$
shows *proj2-Col-coeff* p q (*proj2-abs* ($i *_R \text{proj2-rep } p + j *_R \text{proj2-rep } q$))
 $= i/j$
(is *proj2-Col-coeff* p q $?r = i/j$)
 $\langle \text{proof} \rangle$

lemma *proj2-set-Col-coeff*:
assumes *proj2-set-Col* S **and** $\{p, q, r\} \subseteq S$ **and** $p \neq q$ **and** $r \neq p$
shows $r = \text{proj2-abs} (\text{proj2-Col-coeff } p$ q $r *_R \text{proj2-rep } p + \text{proj2-rep } q)$
(is $r = \text{proj2-abs} (?i *_R ?u + ?v)$)
 $\langle \text{proof} \rangle$

lemma *cross-ratio-abs*:
fixes u $v :: \text{real}^3$ **and** i j k $l :: \text{real}$
assumes $u \neq 0$ **and** $v \neq 0$ **and** *proj2-abs* $u \neq \text{proj2-abs } v$
and $j \neq 0$ **and** $l \neq 0$
shows *cross-ratio* (*proj2-abs* u) (*proj2-abs* v)
(*proj2-abs* ($i *_R u + j *_R v$))
(*proj2-abs* ($k *_R u + l *_R v$))
 $= j * k / (i * l)$
(is *cross-ratio* $?p$ $?q$ $?r$ $?s = -$)
 $\langle \text{proof} \rangle$

lemma *cross-ratio-abs2*:

assumes $p \neq q$

shows *cross-ratio* $p q$

(*proj2-abs* ($i *_R$ *proj2-rep* $p +$ *proj2-rep* q))

(*proj2-abs* ($j *_R$ *proj2-rep* $p +$ *proj2-rep* q))

$= j/i$

(**is** *cross-ratio* $p q$? r ? $s = -$)

<proof>

lemma *cross-ratio-correct-cltn2*:

assumes *cross-ratio-correct* $p q r s$

shows *cross-ratio-correct* (*apply-cltn2* $p C$) (*apply-cltn2* $q C$)

(*apply-cltn2* $r C$) (*apply-cltn2* $s C$)

(**is** *cross-ratio-correct* ? pC ? qC ? rC ? sC)

<proof>

lemma *cross-ratio-cltn2*:

assumes *proj2-set-Col* $\{p,q,r,s\}$ **and** $p \neq q$ **and** $r \neq p$ **and** $s \neq p$

shows *cross-ratio* (*apply-cltn2* $p C$) (*apply-cltn2* $q C$)

(*apply-cltn2* $r C$) (*apply-cltn2* $s C$)

$=$ *cross-ratio* $p q r s$

(**is** *cross-ratio* ? pC ? qC ? rC ? $sC = -$)

<proof>

lemma *cross-ratio-unique*:

assumes *cross-ratio-correct* $p q r s$ **and** *cross-ratio-correct* $p q r t$

and *cross-ratio* $p q r s =$ *cross-ratio* $p q r t$

shows $s = t$

<proof>

lemma *cltn2-three-point-line*:

assumes $p \neq q$ **and** $r \neq p$ **and** $r \neq q$

and *proj2-incident* $p l$ **and** *proj2-incident* $q l$ **and** *proj2-incident* $r l$

and *apply-cltn2* $p C = p$ **and** *apply-cltn2* $q C = q$ **and** *apply-cltn2* $r C = r$

and *proj2-incident* $s l$

shows *apply-cltn2* $s C = s$ (**is** ? $sC = s$)

<proof>

lemma *cross-ratio-equal-cltn2*:

assumes *cross-ratio-correct* $p q r s$

and *cross-ratio-correct* (*apply-cltn2* $p C$) (*apply-cltn2* $q C$)

(*apply-cltn2* $r C$) t

(**is** *cross-ratio-correct* ? pC ? qC ? rC t)

and *cross-ratio* (*apply-cltn2* $p C$) (*apply-cltn2* $q C$) (*apply-cltn2* $r C$) t

$=$ *cross-ratio* $p q r s$

shows $t =$ *apply-cltn2* $s C$ (**is** $t =$? sC)

<proof>

lemma *proj2-Col-distinct-coeff-non-zero*:

assumes *proj2-Col* $p\ q\ r$ **and** $p \neq q$ **and** $r \neq p$ **and** $r \neq q$
shows *proj2-Col-coeff* $p\ q\ r \neq 0$
 ⟨*proof*⟩

lemma *cross-ratio-product*:
assumes *proj2-Col* $p\ q\ s$ **and** $p \neq q$ **and** $s \neq p$ **and** $s \neq q$
shows *cross-ratio* $p\ q\ r\ s * \text{cross-ratio } p\ q\ s\ t = \text{cross-ratio } p\ q\ r\ t$
 ⟨*proof*⟩

lemma *cross-ratio-equal-1*:
assumes *proj2-Col* $p\ q\ r$ **and** $p \neq q$ **and** $r \neq p$ **and** $r \neq q$
shows *cross-ratio* $p\ q\ r\ r = 1$
 ⟨*proof*⟩

lemma *cross-ratio-1-equal*:
assumes *cross-ratio-correct* $p\ q\ r\ s$ **and** *cross-ratio* $p\ q\ r\ s = 1$
shows $r = s$
 ⟨*proof*⟩

lemma *cross-ratio-swap-34*:
shows *cross-ratio* $p\ q\ s\ r = 1 / (\text{cross-ratio } p\ q\ r\ s)$
 ⟨*proof*⟩

lemma *cross-ratio-swap-13-24*:
assumes *cross-ratio-correct* $p\ q\ r\ s$ **and** $r \neq s$
shows *cross-ratio* $r\ s\ p\ q = \text{cross-ratio } p\ q\ r\ s$
 ⟨*proof*⟩

lemma *cross-ratio-swap-12*:
assumes *cross-ratio-correct* $p\ q\ r\ s$ **and** *cross-ratio-correct* $q\ p\ r\ s$
shows *cross-ratio* $q\ p\ r\ s = 1 / (\text{cross-ratio } p\ q\ r\ s)$
 ⟨*proof*⟩

7.6 Cartesian subspace of the real projective plane

definition *vector2-append1* :: $\text{real}^2 \Rightarrow \text{real}^3$ **where**
vector2-append1 $v = \text{vector } [v\$1, v\$2, 1]$

lemma *vector2-append1-non-zero*: *vector2-append1* $v \neq 0$
 ⟨*proof*⟩

definition *proj2-pt* :: $\text{real}^2 \Rightarrow \text{proj2}$ **where**
proj2-pt $v \triangleq \text{proj2-abs } (\text{vector2-append1 } v)$

lemma *proj2-pt-scalar*:
 $\exists c. c \neq 0 \wedge \text{proj2-rep } (\text{proj2-pt } v) = c *_{\mathbb{R}} \text{vector2-append1 } v$
 ⟨*proof*⟩

abbreviation *z-non-zero* :: $\text{proj2} \Rightarrow \text{bool}$ **where**

$z\text{-non-zero } p \triangleq (\text{proj2-rep } p)\$3 \neq 0$

definition $\text{cart2-pt} :: \text{proj2} \Rightarrow \text{real}^2$ **where**

$\text{cart2-pt } p \triangleq$
 $\text{vector } [(\text{proj2-rep } p)\$1 / (\text{proj2-rep } p)\$3, (\text{proj2-rep } p)\$2 / (\text{proj2-rep } p)\$3]$

definition $\text{cart2-append1} :: \text{proj2} \Rightarrow \text{real}^3$ **where**

$\text{cart2-append1 } p \triangleq (1 / ((\text{proj2-rep } p)\$3)) *_{\mathbb{R}} \text{proj2-rep } p$

lemma cart2-append1-z :

assumes $z\text{-non-zero } p$
shows $(\text{cart2-append1 } p)\$3 = 1$
 $\langle \text{proof} \rangle$

lemma $\text{cart2-append1-non-zero}$:

assumes $z\text{-non-zero } p$
shows $\text{cart2-append1 } p \neq 0$
 $\langle \text{proof} \rangle$

lemma $\text{proj2-rep-cart2-append1}$:

assumes $z\text{-non-zero } p$
shows $\text{proj2-rep } p = ((\text{proj2-rep } p)\$3) *_{\mathbb{R}} \text{cart2-append1 } p$
 $\langle \text{proof} \rangle$

lemma $\text{proj2-abs-cart2-append1}$:

assumes $z\text{-non-zero } p$
shows $\text{proj2-abs } (\text{cart2-append1 } p) = p$
 $\langle \text{proof} \rangle$

lemma cart2-append1-inj :

assumes $z\text{-non-zero } p$ **and** $\text{cart2-append1 } p = \text{cart2-append1 } q$
shows $p = q$
 $\langle \text{proof} \rangle$

lemma cart2-append1 :

assumes $z\text{-non-zero } p$
shows $\text{vector2-append1 } (\text{cart2-pt } p) = \text{cart2-append1 } p$
 $\langle \text{proof} \rangle$

lemma cart2-proj2 : $\text{cart2-pt } (\text{proj2-pt } v) = v$

$\langle \text{proof} \rangle$

lemma $z\text{-non-zero-proj2-pt}$: $z\text{-non-zero } (\text{proj2-pt } v)$

$\langle \text{proof} \rangle$

lemma $\text{cart2-append1-proj2}$: $\text{cart2-append1 } (\text{proj2-pt } v) = \text{vector2-append1 } v$

$\langle \text{proof} \rangle$

lemma proj2-pt-inj : $\text{inj } \text{proj2-pt}$

<proof>

lemma *proj2-cart2*:

assumes *z-non-zero p*

shows *proj2-pt (cart2-pt p) = p*

<proof>

lemma *cart2-injective*:

assumes *z-non-zero p and z-non-zero q and cart2-pt p = cart2-pt q*

shows *p = q*

<proof>

lemma *proj2-Col-iff-euclid*:

proj2-Col (proj2-pt a) (proj2-pt b) (proj2-pt c) \longleftrightarrow real-euclid.Col a b c

(is *proj2-Col ?p ?q ?r \longleftrightarrow -)*

<proof>

lemma *proj2-Col-iff-euclid-cart2*:

assumes *z-non-zero p and z-non-zero q and z-non-zero r*

shows

proj2-Col p q r \longleftrightarrow real-euclid.Col (cart2-pt p) (cart2-pt q) (cart2-pt r)

(is *- \longleftrightarrow real-euclid.Col ?a ?b ?c)*

<proof>

lemma *euclid-Col-cart2-incident*:

assumes *z-non-zero p and z-non-zero q and z-non-zero r and $p \neq q$*

and *proj2-incident p l and proj2-incident q l*

and *real-euclid.Col (cart2-pt p) (cart2-pt q) (cart2-pt r)*

(is *real-euclid.Col ?cp ?cq ?cr)*

shows *proj2-incident r l*

<proof>

lemma *euclid-B-cart2-common-line*:

assumes *z-non-zero p and z-non-zero q and z-non-zero r*

and *$B_{\mathbf{R}}$ (cart2-pt p) (cart2-pt q) (cart2-pt r)*

(is *$B_{\mathbf{R}}$?cp ?cq ?cr)*

shows $\exists l. \text{proj2-incident } p \ l \wedge \text{proj2-incident } q \ l \wedge \text{proj2-incident } r \ l$

<proof>

lemma *cart2-append1-between*:

assumes *z-non-zero p and z-non-zero q and z-non-zero r*

shows *$B_{\mathbf{R}}$ (cart2-pt p) (cart2-pt q) (cart2-pt r)*

$\longleftrightarrow (\exists k \geq 0. k \leq 1$

$\wedge \text{cart2-append1 } q = k *_{\mathbf{R}} \text{cart2-append1 } r + (1 - k) *_{\mathbf{R}} \text{cart2-append1 } p)$

<proof>

lemma *cart2-append1-between-right-strict*:

assumes *z-non-zero p and z-non-zero q and z-non-zero r*

and *$B_{\mathbf{R}}$ (cart2-pt p) (cart2-pt q) (cart2-pt r) and $q \neq r$*

shows $\exists k \geq 0. k < 1$
 $\wedge \text{cart2-append1 } q = k *_R \text{cart2-append1 } r + (1 - k) *_R \text{cart2-append1 } p$
 $\langle \text{proof} \rangle$

lemma *cart2-append1-between-strict*:

assumes *z-non-zero p* **and** *z-non-zero q* **and** *z-non-zero r*
and $B_{\mathbb{R}} (\text{cart2-pt } p) (\text{cart2-pt } q) (\text{cart2-pt } r)$ **and** $q \neq p$ **and** $q \neq r$
shows $\exists k > 0. k < 1$
 $\wedge \text{cart2-append1 } q = k *_R \text{cart2-append1 } r + (1 - k) *_R \text{cart2-append1 } p$
 $\langle \text{proof} \rangle$

end

8 The hyperbolic plane and Tarski's axioms

theory *Hyperbolic-Tarski*

imports *Euclid-Tarski*

Projective

HOL-Library.Quadratic-Discriminant

begin

8.1 Characterizing a specific conic in the projective plane

definition $M :: \text{real}^{\wedge 3 \times 3}$ **where**

$M \triangleq \text{vector } [$
 $\text{vector } [1, 0, 0],$
 $\text{vector } [0, 1, 0],$
 $\text{vector } [0, 0, -1]]$

lemma *M-symmatrix: symmatrix M*

$\langle \text{proof} \rangle$

lemma *M-self-inverse: M ** M = mat 1*

$\langle \text{proof} \rangle$

lemma *M-invertible: invertible M*

$\langle \text{proof} \rangle$

definition *polar :: proj2 \Rightarrow proj2-line* **where**

$\text{polar } p \triangleq \text{proj2-line-abs } (M *_v \text{proj2-rep } p)$

definition *pole :: proj2-line \Rightarrow proj2* **where**

$\text{pole } l \triangleq \text{proj2-abs } (M *_v \text{proj2-line-rep } l)$

lemma *polar-abs:*

assumes $v \neq 0$

shows $\text{polar } (\text{proj2-abs } v) = \text{proj2-line-abs } (M *_v v)$

$\langle \text{proof} \rangle$

lemma *pole-abs*:

assumes $v \neq 0$

shows $\text{pole } (\text{proj2-line-abs } v) = \text{proj2-abs } (M *v v)$

$\langle \text{proof} \rangle$

lemma *polar-rep-non-zero*: $M *v \text{proj2-rep } p \neq 0$

$\langle \text{proof} \rangle$

lemma *pole-polar*: $\text{pole } (\text{polar } p) = p$

$\langle \text{proof} \rangle$

lemma *pole-rep-non-zero*: $M *v \text{proj2-line-rep } l \neq 0$

$\langle \text{proof} \rangle$

lemma *polar-pole*: $\text{polar } (\text{pole } l) = l$

$\langle \text{proof} \rangle$

lemma *polar-inj*:

assumes $\text{polar } p = \text{polar } q$

shows $p = q$

$\langle \text{proof} \rangle$

definition *conic-sgn* :: $\text{proj2} \Rightarrow \text{real}$ **where**

$\text{conic-sgn } p \triangleq \text{sgn } (\text{proj2-rep } p \cdot (M *v \text{proj2-rep } p))$

lemma *conic-sgn-abs*:

assumes $v \neq 0$

shows $\text{conic-sgn } (\text{proj2-abs } v) = \text{sgn } (v \cdot (M *v v))$

$\langle \text{proof} \rangle$

lemma *sgn-conic-sgn*: $\text{sgn } (\text{conic-sgn } p) = \text{conic-sgn } p$

$\langle \text{proof} \rangle$

definition *S* :: proj2 set **where**

$S \triangleq \{p. \text{conic-sgn } p = 0\}$

definition *K2* :: proj2 set **where**

$K2 \triangleq \{p. \text{conic-sgn } p < 0\}$

lemma *S-K2-empty*: $S \cap K2 = \{\}$

$\langle \text{proof} \rangle$

lemma *K2-abs*:

assumes $v \neq 0$

shows $\text{proj2-abs } v \in K2 \iff v \cdot (M *v v) < 0$

$\langle \text{proof} \rangle$

definition *K2-centre* = $\text{proj2-abs } (\text{vector } [0,0,1])$

lemma *K2-centre-non-zero*: vector $[0,0,1] \neq (0 :: \text{real}^3)$
<proof>

lemma *K2-centre-in-K2*: $K2\text{-centre} \in K2$
<proof>

lemma *K2-imp-M-neg*:
assumes $v \neq 0$ and $\text{proj2-abs } v \in K2$
shows $v \cdot (M *v v) < 0$
<proof>

lemma *M-neg-imp-z-squared-big*:
assumes $v \cdot (M *v v) < 0$
shows $(v\$3)^2 > (v\$1)^2 + (v\$2)^2$
<proof>

lemma *M-neg-imp-z-non-zero*:
assumes $v \cdot (M *v v) < 0$
shows $v\$3 \neq 0$
<proof>

lemma *M-neg-imp-K2*:
assumes $v \cdot (M *v v) < 0$
shows $\text{proj2-abs } v \in K2$
<proof>

lemma *M-reverse*: $a \cdot (M *v b) = b \cdot (M *v a)$
<proof>

lemma *S-abs*:
assumes $v \neq 0$
shows $\text{proj2-abs } v \in S \iff v \cdot (M *v v) = 0$
<proof>

lemma *S-alt-def*: $p \in S \iff \text{proj2-rep } p \cdot (M *v \text{proj2-rep } p) = 0$
<proof>

lemma *incident-polar*:
 $\text{proj2-incident } p (\text{polar } q) \iff \text{proj2-rep } p \cdot (M *v \text{proj2-rep } q) = 0$
<proof>

lemma *incident-own-polar-in-S*: $\text{proj2-incident } p (\text{polar } p) \iff p \in S$
<proof>

lemma *incident-polar-swap*:
assumes $\text{proj2-incident } p (\text{polar } q)$
shows $\text{proj2-incident } q (\text{polar } p)$
<proof>

lemma *incident-pole-polar*:
assumes *proj2-incident p l*
shows *proj2-incident (pole l) (polar p)*
 \langle *proof* \rangle

definition *z-zero* :: *proj2-line* **where**
z-zero \triangleq *proj2-line-abs (vector [0,0,1])*

lemma *z-zero*:
assumes $(\text{proj2-rep } p)\$3 = 0$
shows *proj2-incident p z-zero*
 \langle *proof* \rangle

lemma *z-zero-conic-sgn-1*:
assumes *proj2-incident p z-zero*
shows *conic-sgn p = 1*
 \langle *proof* \rangle

lemma *conic-sgn-not-1-z-non-zero*:
assumes *conic-sgn p \neq 1*
shows *z-non-zero p*
 \langle *proof* \rangle

lemma *z-zero-not-in-S*:
assumes *proj2-incident p z-zero*
shows $p \notin S$
 \langle *proof* \rangle

lemma *line-incident-point-not-in-S*: $\exists p. p \notin S \wedge \text{proj2-incident } p \ l$
 \langle *proof* \rangle

lemma *apply-cltn2-abs-abs-in-S*:
assumes $v \neq 0$ **and** *invertible J*
shows *apply-cltn2 (proj2-abs v) (cltn2-abs J) \in S*
 $\longleftrightarrow v \cdot (J ** M ** \text{transpose } J * v) = 0$
 \langle *proof* \rangle

lemma *apply-cltn2-right-abs-in-S*:
assumes *invertible J*
shows *apply-cltn2 p (cltn2-abs J) \in S*
 $\longleftrightarrow (\text{proj2-rep } p) \cdot (J ** M ** \text{transpose } J * v (\text{proj2-rep } p)) = 0$
 \langle *proof* \rangle

lemma *apply-cltn2-abs-in-S*:
assumes $v \neq 0$
shows *apply-cltn2 (proj2-abs v) C \in S*
 $\longleftrightarrow v \cdot (\text{cltn2-rep } C ** M ** \text{transpose } (\text{cltn2-rep } C) * v) = 0$
 \langle *proof* \rangle

lemma *apply-cltn2-in-S*:

apply-cltn2 p C ∈ S

$\longleftrightarrow \text{proj2-rep } p \cdot (\text{cltn2-rep } C ** M ** \text{transpose } (\text{cltn2-rep } C) *v \text{proj2-rep } p)$
 $= 0$

<proof>

lemma *norm-M*: $(\text{vector2-append1 } v) \cdot (M *v \text{vector2-append1 } v) = (\text{norm } v)^2 - 1$

<proof>

8.2 Some specific points and lines of the projective plane

definition *east* = *proj2-abs* (*vector* [1,0,1])

definition *west* = *proj2-abs* (*vector* [-1,0,1])

definition *north* = *proj2-abs* (*vector* [0,1,1])

definition *south* = *proj2-abs* (*vector* [0,-1,1])

definition *far-north* = *proj2-abs* (*vector* [0,1,0])

lemmas *compass-defs* = *east-def west-def north-def south-def*

lemma *compass-non-zero*:

shows *vector* [1,0,1] $\neq (0 :: \text{real}^3)$

and *vector* [-1,0,1] $\neq (0 :: \text{real}^3)$

and *vector* [0,1,1] $\neq (0 :: \text{real}^3)$

and *vector* [0,-1,1] $\neq (0 :: \text{real}^3)$

and *vector* [0,1,0] $\neq (0 :: \text{real}^3)$

and *vector* [1,0,0] $\neq (0 :: \text{real}^3)$

<proof>

lemma *east-west-distinct*: *east* \neq *west*

<proof>

lemma *north-south-distinct*: *north* \neq *south*

<proof>

lemma *north-not-east-or-west*: *north* $\notin \{\text{east}, \text{west}\}$

<proof>

lemma *compass-in-S*:

shows *east* $\in S$ **and** *west* $\in S$ **and** *north* $\in S$ **and** *south* $\in S$

<proof>

lemma *east-west-tangents*:

shows *polar east* = *proj2-line-abs* (*vector* [-1,0,1])

and *polar west* = *proj2-line-abs* (*vector* [1,0,1])

<proof>

lemma *east-west-tangents-distinct*: *polar east* \neq *polar west*

<proof>

lemma *east-west-tangents-incident-far-north:*

shows *proj2-incident far-north (polar east)*

and *proj2-incident far-north (polar west)*

<proof>

lemma *east-west-tangents-far-north:*

proj2-intersection (polar east) (polar west) = far-north

<proof>

instantiation *proj2 :: zero*

begin

definition *proj2-zero-def: 0 = proj2-pt 0*

instance *<proof>*

end

definition *equator \triangleq proj2-line-abs (vector [0,1,0])*

definition *meridian \triangleq proj2-line-abs (vector [1,0,0])*

lemma *equator-meridian-distinct: equator \neq meridian*

<proof>

lemma *east-west-on-equator:*

shows *proj2-incident east equator and proj2-incident west equator*

<proof>

lemma *north-far-north-distinct: north \neq far-north*

<proof>

lemma *north-south-far-north-on-meridian:*

shows *proj2-incident north meridian and proj2-incident south meridian*

and *proj2-incident far-north meridian*

<proof>

lemma *K2-centre-on-equator-meridian:*

shows *proj2-incident K2-centre equator*

and *proj2-incident K2-centre meridian*

<proof>

lemma *on-equator-meridian-is-K2-centre:*

assumes *proj2-incident a equator and proj2-incident a meridian*

shows *a = K2-centre*

<proof>

definition *rep-equator-reflect \triangleq vector [*

vector [1, 0,0],

vector [0,-1,0],

vector [0, 0,1]] :: real³

definition *rep-meridian-reflect \triangleq vector [*

vector $[-1,0,0]$,
vector $[0,1,0]$,
vector $[0,0,1]$:: real^3

definition *equator-reflect* \triangleq *cltn2-abs rep-equator-reflect*
definition *meridian-reflect* \triangleq *cltn2-abs rep-meridian-reflect*

lemmas *compass-reflect-defs* = *equator-reflect-def meridian-reflect-def rep-equator-reflect-def rep-meridian-reflect-def*

lemma *compass-reflect-self-inverse*:
shows *rep-equator-reflect ** rep-equator-reflect* = *mat 1*
and *rep-meridian-reflect ** rep-meridian-reflect* = *mat 1*
<proof>

lemma *compass-reflect-invertible*:
shows *invertible rep-equator-reflect and invertible rep-meridian-reflect*
<proof>

lemma *compass-reflect-compass*:
shows *apply-cltn2 east meridian-reflect* = *west*
and *apply-cltn2 west meridian-reflect* = *east*
and *apply-cltn2 north meridian-reflect* = *north*
and *apply-cltn2 south meridian-reflect* = *south*
and *apply-cltn2 K2-centre meridian-reflect* = *K2-centre*
and *apply-cltn2 east equator-reflect* = *east*
and *apply-cltn2 west equator-reflect* = *west*
and *apply-cltn2 north equator-reflect* = *south*
and *apply-cltn2 south equator-reflect* = *north*
and *apply-cltn2 K2-centre equator-reflect* = *K2-centre*
<proof>

lemma *on-equator-rep*:
assumes *z-non-zero a and proj2-incident a equator*
shows $\exists x. a = \text{proj2-abs } (\text{vector } [x,0,1])$
<proof>

lemma *on-meridian-rep*:
assumes *z-non-zero a and proj2-incident a meridian*
shows $\exists y. a = \text{proj2-abs } (\text{vector } [0,y,1])$
<proof>

8.3 Definition of the Klein–Beltrami model of the hyperbolic plane

abbreviation *hyp2* == *K2*

typedef *hyp2* = *K2*
<proof>

definition *hyp2-rep* :: $\text{hyp2} \Rightarrow \text{real}^2$ **where**
 $\text{hyp2-rep } p \triangleq \text{cart2-pt } (\text{Rep-hyp2 } p)$

definition *hyp2-abs* :: $\text{real}^2 \Rightarrow \text{hyp2}$ **where**
 $\text{hyp2-abs } v = \text{Abs-hyp2 } (\text{proj2-pt } v)$

lemma *norm-lt-1-iff-in-hyp2*:
shows $\text{norm } v < 1 \longleftrightarrow \text{proj2-pt } v \in \text{hyp2}$
(*proof*)

lemma *norm-eq-1-iff-in-S*:
shows $\text{norm } v = 1 \longleftrightarrow \text{proj2-pt } v \in S$
(*proof*)

lemma *norm-le-1-iff-in-hyp2-S*:
 $\text{norm } v \leq 1 \longleftrightarrow \text{proj2-pt } v \in \text{hyp2} \cup S$
(*proof*)

lemma *proj2-pt-hyp2-rep*: $\text{proj2-pt } (\text{hyp2-rep } p) = \text{Rep-hyp2 } p$
(*proof*)

lemma *hyp2-rep-abs*:
assumes $\text{norm } v < 1$
shows $\text{hyp2-rep } (\text{hyp2-abs } v) = v$
(*proof*)

lemma *hyp2-abs-rep*: $\text{hyp2-abs } (\text{hyp2-rep } p) = p$
(*proof*)

lemma *norm-hyp2-rep-lt-1*: $\text{norm } (\text{hyp2-rep } p) < 1$
(*proof*)

lemma *hyp2-S-z-non-zero*:
assumes $p \in \text{hyp2} \cup S$
shows $z\text{-non-zero } p$
(*proof*)

lemma *hyp2-S-not-equal*:
assumes $a \in \text{hyp2}$ **and** $p \in S$
shows $a \neq p$
(*proof*)

lemma *hyp2-S-cart2-inj*:
assumes $p \in \text{hyp2} \cup S$ **and** $q \in \text{hyp2} \cup S$ **and** $\text{cart2-pt } p = \text{cart2-pt } q$
shows $p = q$
(*proof*)

lemma *on-equator-in-hyp2-rep*:
assumes $a \in \text{hyp2}$ **and** $\text{proj2-incident } a$ *equator*

shows $\exists x. |x| < 1 \wedge a = \text{proj2-abs (vector [x,0,1])}$
(proof)

lemma *on-meridian-in-hyp2-rep*:
assumes $a \in \text{hyp2}$ **and** *proj2-incident a meridian*
shows $\exists y. |y| < 1 \wedge a = \text{proj2-abs (vector [0,y,1])}$
(proof)

definition *hyp2-cltn2* :: $\text{hyp2} \Rightarrow \text{cltn2} \Rightarrow \text{hyp2}$ **where**
 $\text{hyp2-cltn2 } p \ A \triangleq \text{Abs-hyp2 (apply-cltn2 (Rep-hyp2 } p) \ A)$

definition *is-K2-isometry* :: $\text{cltn2} \Rightarrow \text{bool}$ **where**
 $\text{is-K2-isometry } J \triangleq (\forall p. \text{apply-cltn2 } p \ J \in S \longleftrightarrow p \in S)$

lemma *cltn2-id-is-K2-isometry*: *is-K2-isometry cltn2-id*
(proof)

lemma *J-M-J-transpose-K2-isometry*:
assumes $k \neq 0$
and $\text{repJ} \ ** \ M \ ** \ \text{transpose repJ} = k \ *_{\mathbb{R}} \ M$ (**is** $?N = -$)
shows *is-K2-isometry (cltn2-abs repJ)* (**is** *is-K2-isometry ?J*)
(proof)

lemma *equator-reflect-K2-isometry*:
shows *is-K2-isometry equator-reflect*
(proof)

lemma *meridian-reflect-K2-isometry*:
shows *is-K2-isometry meridian-reflect*
(proof)

lemma *cltn2-compose-is-K2-isometry*:
assumes *is-K2-isometry H* **and** *is-K2-isometry J*
shows *is-K2-isometry (cltn2-compose H J)*
(proof)

lemma *cltn2-inverse-is-K2-isometry*:
assumes *is-K2-isometry J*
shows *is-K2-isometry (cltn2-inverse J)*
(proof)

interpretation *K2-isometry-subgroup*: *subgroup*
Collect is-K2-isometry
(|carrier = UNIV, mult = cltn2-compose, one = cltn2-id|)
(proof)

interpretation *K2-isometry*: *group*
(|carrier = Collect is-K2-isometry, mult = cltn2-compose, one = cltn2-id|)
(proof)

lemma *K2-isometry-inverse-inv* [simp]:
assumes *is-K2-isometry* J
shows $\text{inv}(|\text{carrier} = \text{Collect } \text{is-K2-isometry}, \text{mult} = \text{cltn2-compose}, \text{one} = \text{cltn2-id}|)$
 J
 $= \text{cltn2-inverse } J$
 $\langle \text{proof} \rangle$

definition *real-hyp2-C* :: [*hyp2*, *hyp2*, *hyp2*, *hyp2*] \Rightarrow *bool*
 $(\langle - \equiv_K - \rangle [99,99,99,99] 50)$ **where**
 $p \ q \equiv_K \ r \ s \triangleq$
 $(\exists A. \text{is-K2-isometry } A \wedge \text{hyp2-cltn2 } p \ A = r \wedge \text{hyp2-cltn2 } q \ A = s)$

definition *real-hyp2-B* :: [*hyp2*, *hyp2*, *hyp2*] \Rightarrow *bool*
 $(\langle B_K - - \rangle [99,99,99] 50)$ **where**
 $B_K \ p \ q \ r \triangleq B_{\mathbf{R}} (\text{hyp2-rep } p) (\text{hyp2-rep } q) (\text{hyp2-rep } r)$

8.4 K -isometries map the interior of the conic to itself

lemma *collinear-quadratic*:
assumes $t = i *_R a + r$
shows $t \cdot (M *v t) =$
 $(a \cdot (M *v a)) * i^2 + 2 * (a \cdot (M *v r)) * i + r \cdot (M *v r)$
 $\langle \text{proof} \rangle$

lemma *S-quadratic'*:
assumes $p \neq 0$ **and** $q \neq 0$ **and** $\text{proj2-abs } p \neq \text{proj2-abs } q$
shows $\text{proj2-abs } (k *_R p + q) \in S$
 $\longleftrightarrow p \cdot (M *v p) * k^2 + p \cdot (M *v q) * 2 * k + q \cdot (M *v q) = 0$
 $\langle \text{proof} \rangle$

lemma *S-quadratic*:
assumes $p \neq q$ **and** $r = \text{proj2-abs } (k *_R \text{proj2-rep } p + \text{proj2-rep } q)$
shows $r \in S$
 $\longleftrightarrow \text{proj2-rep } p \cdot (M *v \text{proj2-rep } p) * k^2$
 $\quad + \text{proj2-rep } p \cdot (M *v \text{proj2-rep } q) * 2 * k$
 $\quad + \text{proj2-rep } q \cdot (M *v \text{proj2-rep } q)$
 $= 0$
 $\langle \text{proof} \rangle$

definition *quarter-discrim* :: $\text{real}^3 \Rightarrow \text{real}^3 \Rightarrow \text{real}$ **where**
 $\text{quarter-discrim } p \ q \triangleq (p \cdot (M *v q))^2 - p \cdot (M *v p) * (q \cdot (M *v q))$

lemma *quarter-discrim-invariant*:
assumes $t = i *_R a + r$
shows $\text{quarter-discrim } a \ t = \text{quarter-discrim } a \ r$
 $\langle \text{proof} \rangle$

lemma *quarter-discrim-positive*:

assumes $p \neq 0$ **and** $q \neq 0$ **and** $\text{proj2-abs } p \neq \text{proj2-abs } q$ (**is** $?pp \neq ?pq$)
and $\text{proj2-abs } p \in K2$
shows $\text{quarter-discrim } p \ q > 0$
 <proof>

lemma *quarter-discrim-self-zero*:
assumes $\text{proj2-abs } a = \text{proj2-abs } b$
shows $\text{quarter-discrim } a \ b = 0$
 <proof>

definition *S-intersection-coeff1* :: $\text{real}^3 \Rightarrow \text{real}^3 \Rightarrow \text{real}$ **where**
 $S\text{-intersection-coeff1 } p \ q$
 $\triangleq (-p \cdot (M *v q) + \text{sqrt } (\text{quarter-discrim } p \ q)) / (p \cdot (M *v p))$

definition *S-intersection-coeff2* :: $\text{real}^3 \Rightarrow \text{real}^3 \Rightarrow \text{real}$ **where**
 $S\text{-intersection-coeff2 } p \ q$
 $\triangleq (-p \cdot (M *v q) - \text{sqrt } (\text{quarter-discrim } p \ q)) / (p \cdot (M *v p))$

definition *S-intersection1-rep* :: $\text{real}^3 \Rightarrow \text{real}^3 \Rightarrow \text{real}^3$ **where**
 $S\text{-intersection1-rep } p \ q \triangleq (S\text{-intersection-coeff1 } p \ q) *R \ p + q$

definition *S-intersection2-rep* :: $\text{real}^3 \Rightarrow \text{real}^3 \Rightarrow \text{real}^3$ **where**
 $S\text{-intersection2-rep } p \ q \triangleq (S\text{-intersection-coeff2 } p \ q) *R \ p + q$

definition *S-intersection1* :: $\text{real}^3 \Rightarrow \text{real}^3 \Rightarrow \text{proj2}$ **where**
 $S\text{-intersection1 } p \ q \triangleq \text{proj2-abs } (S\text{-intersection1-rep } p \ q)$

definition *S-intersection2* :: $\text{real}^3 \Rightarrow \text{real}^3 \Rightarrow \text{proj2}$ **where**
 $S\text{-intersection2 } p \ q \triangleq \text{proj2-abs } (S\text{-intersection2-rep } p \ q)$

lemmas *S-intersection-coeffs-defs* =
 $S\text{-intersection-coeff1-def } S\text{-intersection-coeff2-def}$

lemmas *S-intersections-defs* =
 $S\text{-intersection1-def } S\text{-intersection2-def}$
 $S\text{-intersection1-rep-def } S\text{-intersection2-rep-def}$

lemma *S-intersection-coeffs-distinct*:
assumes $p \neq 0$ **and** $q \neq 0$ **and** $\text{proj2-abs } p \neq \text{proj2-abs } q$ (**is** $?pp \neq ?pq$)
and $\text{proj2-abs } p \in K2$
shows $S\text{-intersection-coeff1 } p \ q \neq S\text{-intersection-coeff2 } p \ q$
 <proof>

lemma *S-intersections-distinct*:
assumes $p \neq 0$ **and** $q \neq 0$ **and** $\text{proj2-abs } p \neq \text{proj2-abs } q$ (**is** $?pp \neq ?pq$)
and $\text{proj2-abs } p \in K2$
shows $S\text{-intersection1 } p \ q \neq S\text{-intersection2 } p \ q$
 <proof>

lemma *S-intersections-in-S*:

assumes $p \neq 0$ and $q \neq 0$ and $\text{proj2-abs } p \neq \text{proj2-abs } q$ (is ?pp \neq ?pq)

and $\text{proj2-abs } p \in K2$

shows $S\text{-intersection1 } p \ q \in S$ and $S\text{-intersection2 } p \ q \in S$

<proof>

lemma *S-intersections-Col*:

assumes $p \neq 0$ and $q \neq 0$

shows $\text{proj2-Col } (\text{proj2-abs } p) (\text{proj2-abs } q) (S\text{-intersection1 } p \ q)$

(is $\text{proj2-Col } ?pp \ ?pq \ ?pr$)

and $\text{proj2-Col } (\text{proj2-abs } p) (\text{proj2-abs } q) (S\text{-intersection2 } p \ q)$

(is $\text{proj2-Col } ?pp \ ?pq \ ?ps$)

<proof>

lemma *S-intersections-incident*:

assumes $p \neq 0$ and $q \neq 0$ and $\text{proj2-abs } p \neq \text{proj2-abs } q$ (is ?pp \neq ?pq)

and $\text{proj2-incident } (\text{proj2-abs } p) \ l$ and $\text{proj2-incident } (\text{proj2-abs } q) \ l$

shows $\text{proj2-incident } (S\text{-intersection1 } p \ q) \ l$ (is $\text{proj2-incident } ?pr \ l$)

and $\text{proj2-incident } (S\text{-intersection2 } p \ q) \ l$ (is $\text{proj2-incident } ?ps \ l$)

<proof>

lemma *K2-line-intersect-twice*:

assumes $a \in K2$ and $a \neq r$

shows $\exists s \ u. s \neq u \wedge s \in S \wedge u \in S \wedge \text{proj2-Col } a \ r \ s \wedge \text{proj2-Col } a \ r \ u$

<proof>

lemma *point-in-S-polar-is-tangent*:

assumes $p \in S$ and $q \in S$ and $\text{proj2-incident } q \ (\text{polar } p)$

shows $q = p$

<proof>

lemma *line-through-K2-intersect-S-twice*:

assumes $p \in K2$ and $\text{proj2-incident } p \ l$

shows $\exists q \ r. q \neq r \wedge q \in S \wedge r \in S \wedge \text{proj2-incident } q \ l \wedge \text{proj2-incident } r \ l$

<proof>

lemma *line-through-K2-intersect-S-again*:

assumes $p \in K2$ and $\text{proj2-incident } p \ l$

shows $\exists r. r \neq p \wedge r \in S \wedge \text{proj2-incident } r \ l$

<proof>

lemma *line-through-K2-intersect-S*:

assumes $p \in K2$ and $\text{proj2-incident } p \ l$

shows $\exists r. r \in S \wedge \text{proj2-incident } r \ l$

<proof>

lemma *line-intersect-S-at-most-twice*:

$\exists p \ q. \forall r \in S. \text{proj2-incident } r \ l \longrightarrow r = p \vee r = q$

<proof>

lemma *card-line-intersect-S*:

assumes $T \subseteq S$ **and** *proj2-set-Col T*

shows $\text{card } T \leq 2$

<proof>

lemma *line-S-two-intersections-only*:

assumes $p \neq q$ **and** $p \in S$ **and** $q \in S$ **and** $r \in S$

and *proj2-incident p l* **and** *proj2-incident q l* **and** *proj2-incident r l*

shows $r = p \vee r = q$

<proof>

lemma *line-through-K2-intersect-S-exactly-twice*:

assumes $p \in K2$ **and** *proj2-incident p l*

shows $\exists q r. q \neq r \wedge q \in S \wedge r \in S \wedge \text{proj2-incident } q l \wedge \text{proj2-incident } r l$

$\wedge (\forall s \in S. \text{proj2-incident } s l \longrightarrow s = q \vee s = r)$

<proof>

lemma *tangent-not-through-K2*:

assumes $p \in S$ **and** $q \in K2$

shows $\neg \text{proj2-incident } q (\text{polar } p)$

<proof>

lemma *outside-exists-line-not-intersect-S*:

assumes $\text{conic-sgn } p = 1$

shows $\exists l. \text{proj2-incident } p l \wedge (\forall q. \text{proj2-incident } q l \longrightarrow q \notin S)$

<proof>

lemma *lines-through-intersect-S-twice-in-K2*:

assumes $\forall l. \text{proj2-incident } p l$

$\longrightarrow (\exists q r. q \neq r \wedge q \in S \wedge r \in S \wedge \text{proj2-incident } q l \wedge \text{proj2-incident } r l)$

shows $p \in K2$

<proof>

lemma *line-through-hyp2-pole-not-in-hyp2*:

assumes $a \in \text{hyp2}$ **and** *proj2-incident a l*

shows $\text{pole } l \notin \text{hyp2}$

<proof>

lemma *statement60-one-way*:

assumes *is-K2-isometry J* **and** $p \in K2$

shows *apply-cltn2 p J* $\in K2$ (**is** $?p' \in K2$)

<proof>

lemma *is-K2-isometry-hyp2-S*:

assumes $p \in \text{hyp2} \cup S$ **and** *is-K2-isometry J*

shows *apply-cltn2 p J* $\in \text{hyp2} \cup S$

<proof>

lemma *is-K2-isometry-z-non-zero*:
assumes $p \in \text{hyp2} \cup S$ **and** *is-K2-isometry* J
shows *z-non-zero* (*apply-cltn2* p J)
<proof>

lemma *cart2-append1-apply-cltn2*:
assumes $p \in \text{hyp2} \cup S$ **and** *is-K2-isometry* J
shows $\exists k. k \neq 0$
 \wedge *cart2-append1* p v^* *cltn2-rep* $J = k *_R$ *cart2-append1* (*apply-cltn2* p J)
<proof>

8.5 The K -isometries form a group action

lemma *hyp2-cltn2-id [simp]*: *hyp2-cltn2* p *cltn2-id* = p
<proof>

lemma *apply-cltn2-Rep-hyp2*:
assumes *is-K2-isometry* J
shows *apply-cltn2* (*Rep-hyp2* p) $J \in \text{hyp2}$
<proof>

lemma *Rep-hyp2-cltn2*:
assumes *is-K2-isometry* J
shows *Rep-hyp2* (*hyp2-cltn2* p J) = *apply-cltn2* (*Rep-hyp2* p) J
<proof>

lemma *hyp2-cltn2-compose*:
assumes *is-K2-isometry* H
shows *hyp2-cltn2* (*hyp2-cltn2* p H) $J = \text{hyp2-cltn2 } p$ (*cltn2-compose* H J)
<proof>

interpretation *K2-isometry: action*
(|*carrier* = *Collect is-K2-isometry*, *mult* = *cltn2-compose*, *one* = *cltn2-id*)
hyp2-cltn2
<proof>

8.6 The Klein–Beltrami model satisfies Tarski’s first three axioms

lemma *three-in-S-tangent-intersection-no-3-Col*:
assumes $p \in S$ **and** $q \in S$ **and** $r \in S$
and $p \neq q$ **and** $r \notin \{p, q\}$
shows *proj2-no-3-Col* {*proj2-intersection* (*polar* p) (*polar* q), r, p, q }
(**is** *proj2-no-3-Col* { $?s, r, p, q$ })
<proof>

lemma *statement65-special-case*:
assumes $p \in S$ **and** $q \in S$ **and** $r \in S$ **and** $p \neq q$ **and** $r \notin \{p, q\}$
shows $\exists J. \text{is-K2-isometry } J$

\wedge *apply-cltn2 east* $J = p$
 \wedge *apply-cltn2 west* $J = q$
 \wedge *apply-cltn2 north* $J = r$
 \wedge *apply-cltn2 far-north* $J = \text{proj2-intersection (polar } p) \text{ (polar } q)$
 <proof>

lemma *statement66-existence*:

assumes $a1 \in K2$ **and** $a2 \in K2$ **and** $p1 \in S$ **and** $p2 \in S$
shows $\exists J. \text{is-}K2\text{-isometry } J \wedge \text{apply-cltn2 } a1 \ J = a2 \wedge \text{apply-cltn2 } p1 \ J = p2$
 <proof>

lemma *K2-isometry-swap*:

assumes $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$
shows $\exists J. \text{is-}K2\text{-isometry } J \wedge \text{apply-cltn2 } a \ J = b \wedge \text{apply-cltn2 } b \ J = a$
 <proof>

theorem *hyp2-axiom1*: $\forall a \ b. a \ b \equiv_K b \ a$
 <proof>

theorem *hyp2-axiom2*: $\forall a \ b \ p \ q \ r \ s. a \ b \equiv_K p \ q \wedge a \ b \equiv_K r \ s \longrightarrow p \ q \equiv_K r \ s$
 <proof>

theorem *hyp2-axiom3*: $\forall a \ b \ c. a \ b \equiv_K c \ c \longrightarrow a = b$
 <proof>

interpretation *hyp2*: *tarski-first3 real-hyp2-C*
 <proof>

8.7 Some lemmas about betweenness

lemma *S-at-edge*:

assumes $p \in S$ **and** $q \in \text{hyp2} \cup S$ **and** $r \in \text{hyp2} \cup S$ **and** *proj2-Col* $p \ q \ r$
shows $B_{\mathbf{R}} (\text{cart2-pt } p) (\text{cart2-pt } q) (\text{cart2-pt } r)$
 $\vee B_{\mathbf{R}} (\text{cart2-pt } p) (\text{cart2-pt } r) (\text{cart2-pt } q)$
(is $B_{\mathbf{R}} \ ?cp \ ?cq \ ?cr \ \vee \ -)$
 <proof>

lemma *hyp2-in-middle*:

assumes $p \in S$ **and** $q \in S$ **and** $r \in \text{hyp2} \cup S$ **and** *proj2-Col* $p \ q \ r$
and $p \neq q$
shows $B_{\mathbf{R}} (\text{cart2-pt } p) (\text{cart2-pt } r) (\text{cart2-pt } q)$ **(is** $B_{\mathbf{R}} \ ?cp \ ?cr \ ?cq)$
 <proof>

lemma *hyp2-incident-in-middle*:

assumes $p \neq q$ **and** $p \in S$ **and** $q \in S$ **and** $a \in \text{hyp2} \cup S$
and *proj2-incident* $p \ l$ **and** *proj2-incident* $q \ l$ **and** *proj2-incident* $a \ l$
shows $B_{\mathbf{R}} (\text{cart2-pt } p) (\text{cart2-pt } a) (\text{cart2-pt } q)$
 <proof>

lemma *extend-to-S*:

assumes $p \in \text{hyp2} \cup S$ **and** $q \in \text{hyp2} \cup S$
shows $\exists r \in S. B_{\mathbf{R}} (\text{cart2-pt } p) (\text{cart2-pt } q) (\text{cart2-pt } r)$
(is $\exists r \in S. B_{\mathbf{R}} ?cp ?cq (\text{cart2-pt } r)$)

<proof>

definition *endpoint-in-S* :: $\text{proj2} \Rightarrow \text{proj2} \Rightarrow \text{proj2}$ **where**

endpoint-in-S a b
 $\triangleq \epsilon p. p \in S \wedge B_{\mathbf{R}} (\text{cart2-pt } a) (\text{cart2-pt } b) (\text{cart2-pt } p)$

lemma *endpoint-in-S*:

assumes $a \in \text{hyp2} \cup S$ **and** $b \in \text{hyp2} \cup S$
shows *endpoint-in-S* a $b \in S$ **(is** $?p \in S$)
and $B_{\mathbf{R}} (\text{cart2-pt } a) (\text{cart2-pt } b) (\text{cart2-pt } (\text{endpoint-in-S } a \ b))$
(is $B_{\mathbf{R}} ?ca ?cb ?cp$)

<proof>

lemma *endpoint-in-S-swap*:

assumes $a \neq b$ **and** $a \in \text{hyp2} \cup S$ **and** $b \in \text{hyp2} \cup S$
shows *endpoint-in-S* a $b \neq \text{endpoint-in-S } b$ a **(is** $?p \neq ?q$)

<proof>

lemma *endpoint-in-S-incident*:

assumes $a \neq b$ **and** $a \in \text{hyp2} \cup S$ **and** $b \in \text{hyp2} \cup S$
and *proj2-incident* a l **and** *proj2-incident* b l
shows *proj2-incident* (*endpoint-in-S* a b) l **(is** *proj2-incident* $?p$ l)

<proof>

lemma *endpoints-in-S-incident-unique*:

assumes $a \neq b$ **and** $a \in \text{hyp2} \cup S$ **and** $b \in \text{hyp2} \cup S$ **and** $p \in S$
and *proj2-incident* a l **and** *proj2-incident* b l **and** *proj2-incident* p l
shows $p = \text{endpoint-in-S } a$ $b \vee p = \text{endpoint-in-S } b$ a
(is $p = ?q \vee p = ?r$)

<proof>

lemma *endpoint-in-S-unique*:

assumes $a \neq b$ **and** $a \in \text{hyp2} \cup S$ **and** $b \in \text{hyp2} \cup S$ **and** $p \in S$
and $B_{\mathbf{R}} (\text{cart2-pt } a) (\text{cart2-pt } b) (\text{cart2-pt } p)$ **(is** $B_{\mathbf{R}} ?ca ?cb ?cp$)
shows $p = \text{endpoint-in-S } a$ b **(is** $p = ?q$)

<proof>

lemma *between-hyp2-S*:

assumes $p \in \text{hyp2} \cup S$ **and** $r \in \text{hyp2} \cup S$ **and** $k \geq 0$ **and** $k \leq 1$
shows *proj2-pt* ($k *_{\mathbf{R}} (\text{cart2-pt } r) + (1 - k) *_{\mathbf{R}} (\text{cart2-pt } p)$) $\in \text{hyp2} \cup S$
(is *proj2-pt* $?cq \in -$)

<proof>

8.8 The Klein–Beltrami model satisfies axiom 4

definition *expansion-factor* :: *proj2* \Rightarrow *cltn2* \Rightarrow *real* **where**
expansion-factor *p J* \triangleq (*cart2-append1* *p v* cltn2-rep J*)\$3

lemma *expansion-factor*:

assumes $p \in \text{hyp2} \cup S$ **and** *is-K2-isometry* *J*
shows *expansion-factor* *p J* $\neq 0$
and *cart2-append1* *p v* cltn2-rep J*
 $= \text{expansion-factor } p J *_{\mathbb{R}} \text{cart2-append1 } (\text{apply-cltn2 } p J)$

<proof>

lemma *expansion-factor-linear-apply-cltn2*:

assumes $p \in \text{hyp2} \cup S$ **and** $q \in \text{hyp2} \cup S$ **and** $r \in \text{hyp2} \cup S$
and *is-K2-isometry* *J*
and *cart2-pt* $r = k *_{\mathbb{R}} \text{cart2-pt } p + (1 - k) *_{\mathbb{R}} \text{cart2-pt } q$
shows *expansion-factor* $r J *_{\mathbb{R}} \text{cart2-append1 } (\text{apply-cltn2 } r J)$
 $= (k * \text{expansion-factor } p J) *_{\mathbb{R}} \text{cart2-append1 } (\text{apply-cltn2 } p J)$
 $+ ((1 - k) * \text{expansion-factor } q J) *_{\mathbb{R}} \text{cart2-append1 } (\text{apply-cltn2 } q J)$
(is $?er *_{\mathbb{R}} - = (k * ?ep) *_{\mathbb{R}} - + ((1 - k) * ?eq) *_{\mathbb{R}} -$ **)**

<proof>

lemma *expansion-factor-linear*:

assumes $p \in \text{hyp2} \cup S$ **and** $q \in \text{hyp2} \cup S$ **and** $r \in \text{hyp2} \cup S$
and *is-K2-isometry* *J*
and *cart2-pt* $r = k *_{\mathbb{R}} \text{cart2-pt } p + (1 - k) *_{\mathbb{R}} \text{cart2-pt } q$
shows *expansion-factor* $r J$
 $= k * \text{expansion-factor } p J + (1 - k) * \text{expansion-factor } q J$
(is $?er = k * ?ep + (1 - k) * ?eq$ **)**

<proof>

lemma *expansion-factor-sgn-invariant*:

assumes $p \in \text{hyp2} \cup S$ **and** $q \in \text{hyp2} \cup S$ **and** *is-K2-isometry* *J*
shows *sgn* (*expansion-factor* *p J*) = *sgn* (*expansion-factor* *q J*)
(is $\text{sgn } ?ep = \text{sgn } ?eq$ **)**

<proof>

lemma *statement-63*:

assumes $p \in \text{hyp2} \cup S$ **and** $q \in \text{hyp2} \cup S$ **and** $r \in \text{hyp2} \cup S$
and *is-K2-isometry* *J* **and** $B_{\mathbb{R}}$ (*cart2-pt* *p*) (*cart2-pt* *q*) (*cart2-pt* *r*)
shows $B_{\mathbb{R}}$
(*cart2-pt* (*apply-cltn2* *p J*))
(*cart2-pt* (*apply-cltn2* *q J*))
(*cart2-pt* (*apply-cltn2* *r J*))

<proof>

theorem *hyp2-axiom4*: $\forall q a b c. \exists x. B_K q a x \wedge a x \equiv_K b c$

<proof>

8.9 More betweenness theorems

lemma *hyp2-S-points-fix-line*:

assumes $a \in \text{hyp2}$ and $p \in S$ and *is-K2-isometry* J
and *apply-cltn2* $a J = a$ (is ?aJ = a)
and *apply-cltn2* $p J = p$ (is ?pJ = p)
and *proj2-incident* $a l$ and *proj2-incident* $p l$ and *proj2-incident* $b l$
shows *apply-cltn2* $b J = b$ (is ?bJ = b)

<proof>

lemma *K2-isometry-endpoint-in-S*:

assumes $a \neq b$ and $a \in \text{hyp2} \cup S$ and $b \in \text{hyp2} \cup S$ and *is-K2-isometry* J
shows *apply-cltn2* (*endpoint-in-S* $a b$) J
= *endpoint-in-S* (*apply-cltn2* $a J$) (*apply-cltn2* $b J$)
(is ?pJ = *endpoint-in-S* ?aJ ?bJ)

<proof>

lemma *between-endpoint-in-S*:

assumes $a \neq b$ and $b \neq c$
and $a \in \text{hyp2} \cup S$ and $b \in \text{hyp2} \cup S$ and $c \in \text{hyp2} \cup S$
and $B_{\mathbf{R}}$ (*cart2-pt* a) (*cart2-pt* b) (*cart2-pt* c) (is $B_{\mathbf{R}}$?ca ?cb ?cc)
shows *endpoint-in-S* $a b = \text{endpoint-in-S } b c$ (is ?p = ?q)

<proof>

lemma *hyp2-extend-segment-unique*:

assumes $a \neq b$ and $B_K a b c$ and $B_K a b d$ and $b c \equiv_K b d$
shows $c = d$

<proof>

lemma *line-S-match-intersections*:

assumes $p \neq q$ and $r \neq s$ and $p \in S$ and $q \in S$ and $r \in S$ and $s \in S$
and *proj2-set-Col* $\{p, q, r, s\}$
shows $(p = r \wedge q = s) \vee (q = r \wedge p = s)$

<proof>

definition *are-endpoints-in-S* :: [*proj2*, *proj2*, *proj2*, *proj2*] \Rightarrow bool **where**

are-endpoints-in-S $p q a b$
 $\triangleq p \neq q \wedge p \in S \wedge q \in S \wedge a \in \text{hyp2} \wedge b \in \text{hyp2} \wedge \text{proj2-set-Col } \{p, q, a, b\}$

lemma *are-endpoints-in-S'*:

assumes $p \neq q$ and $a \neq b$ and $p \in S$ and $q \in S$ and $a \in \text{hyp2} \cup S$
and $b \in \text{hyp2} \cup S$ and *proj2-set-Col* $\{p, q, a, b\}$
shows $(p = \text{endpoint-in-S } a b \wedge q = \text{endpoint-in-S } b a)$
 $\vee (q = \text{endpoint-in-S } a b \wedge p = \text{endpoint-in-S } b a)$
(is $(p = ?r \wedge q = ?s) \vee (q = ?r \wedge p = ?s)$)

<proof>

lemma *are-endpoints-in-S*:

assumes $a \neq b$ and *are-endpoints-in-S* $p q a b$
shows $(p = \text{endpoint-in-S } a b \wedge q = \text{endpoint-in-S } b a)$

$\vee (q = \text{endpoint-in-}S\ a\ b \wedge p = \text{endpoint-in-}S\ b\ a)$
 $\langle \text{proof} \rangle$

lemma *S-intersections-endpoints-in-S:*

assumes $a \neq 0$ **and** $b \neq 0$ **and** $\text{proj2-abs}\ a \neq \text{proj2-abs}\ b$ (**is** $?pa \neq ?pb$)
and $\text{proj2-abs}\ a \in \text{hyp2}$ **and** $\text{proj2-abs}\ b \in \text{hyp2} \cup S$
shows $(S\text{-intersection1}\ a\ b = \text{endpoint-in-}S\ ?pa\ ?pb$
 $\wedge S\text{-intersection2}\ a\ b = \text{endpoint-in-}S\ ?pb\ ?pa)$
 $\vee (S\text{-intersection2}\ a\ b = \text{endpoint-in-}S\ ?pa\ ?pb$
 $\wedge S\text{-intersection1}\ a\ b = \text{endpoint-in-}S\ ?pb\ ?pa)$
(is $(?pp = ?pr \wedge ?pq = ?ps) \vee (?pq = ?pr \wedge ?pp = ?ps)$)
 $\langle \text{proof} \rangle$

lemma *between-endpoints-in-S:*

assumes $a \neq b$ **and** $a \in \text{hyp2} \cup S$ **and** $b \in \text{hyp2} \cup S$
shows $B_{\mathbb{R}}$
 $(\text{cart2-pt}\ (\text{endpoint-in-}S\ a\ b))\ (\text{cart2-pt}\ a)\ (\text{cart2-pt}\ (\text{endpoint-in-}S\ b\ a))$
(is $B_{\mathbb{R}}\ ?cp\ ?ca\ ?cq)$
 $\langle \text{proof} \rangle$

lemma *S-hyp2-S-cart2-append1:*

assumes $p \neq q$ **and** $p \in S$ **and** $q \in S$ **and** $a \in \text{hyp2}$
and $\text{proj2-incident}\ p\ l$ **and** $\text{proj2-incident}\ q\ l$ **and** $\text{proj2-incident}\ a\ l$
shows $\exists k. k > 0 \wedge k < 1$
 $\wedge \text{cart2-append1}\ a = k *_{\mathbb{R}} \text{cart2-append1}\ q + (1 - k) *_{\mathbb{R}} \text{cart2-append1}\ p$
 $\langle \text{proof} \rangle$

lemma *are-endpoints-in-S-swap-34:*

assumes $\text{are-endpoints-in-}S\ p\ q\ a\ b$
shows $\text{are-endpoints-in-}S\ p\ q\ b\ a$
 $\langle \text{proof} \rangle$

lemma *proj2-set-Col-endpoints-in-S:*

assumes $a \neq b$ **and** $a \in \text{hyp2} \cup S$ **and** $b \in \text{hyp2} \cup S$
shows $\text{proj2-set-Col}\ \{\text{endpoint-in-}S\ a\ b, \text{endpoint-in-}S\ b\ a, a, b\}$
(is $\text{proj2-set-Col}\ \{?p, ?q, a, b\}$)
 $\langle \text{proof} \rangle$

lemma *endpoints-in-S-are-endpoints-in-S:*

assumes $a \neq b$ **and** $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$
shows $\text{are-endpoints-in-}S\ (\text{endpoint-in-}S\ a\ b)\ (\text{endpoint-in-}S\ b\ a)\ a\ b$
(is $\text{are-endpoints-in-}S\ ?p\ ?q\ a\ b)$
 $\langle \text{proof} \rangle$

lemma *endpoint-in-S-S-hyp2-distinct:*

assumes $p \in S$ **and** $a \in \text{hyp2} \cup S$ **and** $p \neq a$
shows $\text{endpoint-in-}S\ p\ a \neq p$
 $\langle \text{proof} \rangle$

lemma *endpoint-in-S-S-strict-hyp2-distinct*:

assumes $p \in S$ **and** $a \in \text{hyp2}$

shows *endpoint-in-S* p $a \neq p$

<proof>

lemma *end-and-opposite-are-endpoints-in-S*:

assumes $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$ **and** $p \in S$

and *proj2-incident* a l **and** *proj2-incident* b l **and** *proj2-incident* p l

shows *are-endpoints-in-S* p (*endpoint-in-S* p b) a b

(**is** *are-endpoints-in-S* p $?q$ a b)

<proof>

lemma *real-hyp2-B-hyp2-cltn2*:

assumes *is-K2-isometry* J **and** B_K a b c

shows B_K (*hyp2-cltn2* a J) (*hyp2-cltn2* b J) (*hyp2-cltn2* c J)

(**is** B_K $?aJ$ $?bJ$ $?cJ$)

<proof>

lemma *real-hyp2-C-hyp2-cltn2*:

assumes *is-K2-isometry* J

shows a $b \equiv_K$ (*hyp2-cltn2* a J) (*hyp2-cltn2* b J) (**is** a $b \equiv_K$ $?aJ$ $?bJ$)

<proof>

8.10 Perpendicularity

definition *M-perp* :: *proj2-line* \Rightarrow *proj2-line* \Rightarrow *bool* **where**

M-perp l $m \triangleq$ *proj2-incident* (*pole* l) m

lemma *M-perp-sym*:

assumes *M-perp* l m

shows *M-perp* m l

<proof>

lemma *M-perp-to-compass*:

assumes *M-perp* l m **and** $a \in \text{hyp2}$ **and** *proj2-incident* a l

and $b \in \text{hyp2}$ **and** *proj2-incident* b m

shows $\exists J$. *is-K2-isometry* J

\wedge *apply-cltn2-line* *equator* $J = l \wedge$ *apply-cltn2-line* *meridian* $J = m$

<proof>

definition *drop-perp* :: *proj2* \Rightarrow *proj2-line* \Rightarrow *proj2-line* **where**

drop-perp p $l \triangleq$ *proj2-line-through* p (*pole* l)

lemma *drop-perp-incident*: *proj2-incident* p (*drop-perp* p l)

<proof>

lemma *drop-perp-perp*: *M-perp* l (*drop-perp* p l)

<proof>

definition *perp-foot* :: *proj2* \Rightarrow *proj2-line* \Rightarrow *proj2* **where**
perp-foot *p l* \triangleq *proj2-intersection l (drop-perp p l)*

lemma *perp-foot-incident*:
shows *proj2-incident (perp-foot p l) l*
and *proj2-incident (perp-foot p l) (drop-perp p l)*
<proof>

lemma *M-perp-hyp2*:
assumes *M-perp l m* **and** *a* \in *hyp2* **and** *proj2-incident a l* **and** *b* \in *hyp2*
and *proj2-incident b m* **and** *proj2-incident c l* **and** *proj2-incident c m*
shows *c* \in *hyp2*
<proof>

lemma *perp-foot-hyp2*:
assumes *a* \in *hyp2* **and** *proj2-incident a l* **and** *b* \in *hyp2*
shows *perp-foot b l* \in *hyp2*
<proof>

definition *perp-up* :: *proj2* \Rightarrow *proj2-line* \Rightarrow *proj2* **where**
perp-up a l
 \triangleq *if proj2-incident a l then* ϵ *p. p* \in *S* \wedge *proj2-incident p (drop-perp a l)*
else endpoint-in-S (perp-foot a l) a

lemma *perp-up-degenerate-in-S-incident*:
assumes *a* \in *hyp2* **and** *proj2-incident a l*
shows *perp-up a l* \in *S* (**is** *?p* \in *S*)
and *proj2-incident (perp-up a l) (drop-perp a l)*
<proof>

lemma *perp-up-non-degenerate-in-S-at-end*:
assumes *a* \in *hyp2* **and** *b* \in *hyp2* **and** *proj2-incident b l*
and \neg *proj2-incident a l*
shows *perp-up a l* \in *S*
and $B_{\mathbb{R}}$ (*cart2-pt (perp-foot a l)*) (*cart2-pt a*) (*cart2-pt (perp-up a l)*)
<proof>

lemma *perp-up-in-S*:
assumes *a* \in *hyp2* **and** *b* \in *hyp2* **and** *proj2-incident b l*
shows *perp-up a l* \in *S*
<proof>

lemma *perp-up-incident*:
assumes *a* \in *hyp2* **and** *b* \in *hyp2* **and** *proj2-incident b l*
shows *proj2-incident (perp-up a l) (drop-perp a l)*
(is *proj2-incident ?p ?m*)
<proof>

lemma *drop-perp-same-line-pole-in-S*:

assumes $\text{drop-perp } p \ l = l$
shows $\text{pole } l \in S$
 <proof>

lemma *hyp2-drop-perp-not-same-line*:
assumes $a \in \text{hyp2}$
shows $\text{drop-perp } a \ l \neq l$
 <proof>

lemma *hyp2-incident-perp-foot-same-point*:
assumes $a \in \text{hyp2}$ **and** $\text{proj2-incident } a \ l$
shows $\text{perp-foot } a \ l = a$
 <proof>

lemma *perp-up-at-end*:
assumes $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$ **and** $\text{proj2-incident } b \ l$
shows $B_{\mathbb{R}} (\text{cart2-pt } (\text{perp-foot } a \ l)) (\text{cart2-pt } a) (\text{cart2-pt } (\text{perp-up } a \ l))$
 <proof>

definition *perp-down* :: $\text{proj2} \Rightarrow \text{proj2-line} \Rightarrow \text{proj2}$ **where**
 $\text{perp-down } a \ l \triangleq \text{endpoint-in-S } (\text{perp-up } a \ l) \ a$

lemma *perp-down-in-S*:
assumes $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$ **and** $\text{proj2-incident } b \ l$
shows $\text{perp-down } a \ l \in S$
 <proof>

lemma *perp-down-incident*:
assumes $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$ **and** $\text{proj2-incident } b \ l$
shows $\text{proj2-incident } (\text{perp-down } a \ l) (\text{drop-perp } a \ l)$
 <proof>

lemma *perp-up-down-distinct*:
assumes $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$ **and** $\text{proj2-incident } b \ l$
shows $\text{perp-up } a \ l \neq \text{perp-down } a \ l$
 <proof>

lemma *perp-up-down-foot-are-endpoints-in-S*:
assumes $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$ **and** $\text{proj2-incident } b \ l$
shows $\text{are-endpoints-in-S } (\text{perp-up } a \ l) (\text{perp-down } a \ l) (\text{perp-foot } a \ l) \ a$
 <proof>

lemma *perp-foot-opposite-endpoint-in-S*:
assumes $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$ **and** $c \in \text{hyp2}$ **and** $a \neq b$
shows
 $\text{endpoint-in-S } (\text{endpoint-in-S } a \ b) (\text{perp-foot } c (\text{proj2-line-through } a \ b))$
 $= \text{endpoint-in-S } b \ a$
 (is $\text{endpoint-in-S } ?p \ ?d = \text{endpoint-in-S } b \ a$)
 <proof>

lemma *endpoints-in-S-perp-foot-are-endpoints-in-S*:
assumes $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$ **and** $c \in \text{hyp2}$ **and** $a \neq b$
and *proj2-incident* a l **and** *proj2-incident* b l
shows *are-endpoints-in-S*
(endpoint-in-S a b) *(endpoint-in-S* b a) a *(perp-foot* c l)
 $\langle \text{proof} \rangle$

definition *right-angle* :: *proj2* \Rightarrow *proj2* \Rightarrow *proj2* \Rightarrow *bool* **where**
right-angle p a q
 $\triangleq p \in S \wedge q \in S \wedge a \in \text{hyp2}$
 $\wedge M\text{-perp}$ (*proj2-line-through* p a) (*proj2-line-through* a q)

lemma *perp-foot-up-right-angle*:
assumes $p \in S$ **and** $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$ **and** *proj2-incident* p l
and *proj2-incident* b l
shows *right-angle* p (*perp-foot* a l) (*perp-up* a l)
 $\langle \text{proof} \rangle$

lemma *M-perp-unique*:
assumes $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$ **and** *proj2-incident* a l
and *proj2-incident* b m **and** *proj2-incident* b n **and** *M-perp* l m
and *M-perp* l n
shows $m = n$
 $\langle \text{proof} \rangle$

lemma *perp-foot-eq-implies-drop-perp-eq*:
assumes $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$ **and** *proj2-incident* a l
and *perp-foot* b l = *perp-foot* c l
shows *drop-perp* b l = *drop-perp* c l
 $\langle \text{proof} \rangle$

lemma *right-angle-to-compass*:
assumes *right-angle* p a q
shows $\exists J. \text{is-K2-isometry } J \wedge \text{apply-cltn2 } p$ J = *east*
 $\wedge \text{apply-cltn2 } a$ J = *K2-centre* $\wedge \text{apply-cltn2 } q$ J = *north*
 $\langle \text{proof} \rangle$

lemma *right-angle-to-right-angle*:
assumes *right-angle* p a q **and** *right-angle* r b s
shows $\exists J. \text{is-K2-isometry } J$
 $\wedge \text{apply-cltn2 } p$ J = r $\wedge \text{apply-cltn2 } a$ J = b $\wedge \text{apply-cltn2 } q$ J = s
 $\langle \text{proof} \rangle$

8.11 Functions of distance

definition *exp-2dist* :: *proj2* \Rightarrow *proj2* \Rightarrow *real* **where**
exp-2dist a b
 $\triangleq \text{if } a = b$

then 1
 else cross-ratio (endpoint-in-S a b) (endpoint-in-S b a) a b

definition *cosh-dist* :: proj2 ⇒ proj2 ⇒ real **where**
cosh-dist a b ≙ (sqrt (exp-2dist a b) + sqrt (1 / (exp-2dist a b))) / 2

lemma *exp-2dist-formula*:
assumes $a \neq 0$ **and** $b \neq 0$ **and** *proj2-abs* a ∈ *hyp2* (**is** ?pa ∈ *hyp2*)
and *proj2-abs* b ∈ *hyp2* (**is** ?pb ∈ *hyp2*)
shows *exp-2dist* (*proj2-abs* a) (*proj2-abs* b)
 = (a · (M *v b) + sqrt (quarter-discrim a b))
 / (a · (M *v b) - sqrt (quarter-discrim a b))
 ∨ *exp-2dist* (*proj2-abs* a) (*proj2-abs* b)
 = (a · (M *v b) - sqrt (quarter-discrim a b))
 / (a · (M *v b) + sqrt (quarter-discrim a b))
(is ?e2d = (?aMb + ?sqd) / (?aMb - ?sqd)
 ∨ ?e2d = (?aMb - ?sqd) / (?aMb + ?sqd))
 ⟨proof⟩

lemma *cosh-dist-formula*:
assumes $a \neq 0$ **and** $b \neq 0$ **and** *proj2-abs* a ∈ *hyp2* (**is** ?pa ∈ *hyp2*)
and *proj2-abs* b ∈ *hyp2* (**is** ?pb ∈ *hyp2*)
shows *cosh-dist* (*proj2-abs* a) (*proj2-abs* b)
 = |a · (M *v b)| / sqrt (a · (M *v a) * (b · (M *v b)))
(is *cosh-dist* ?pa ?pb = |?aMb| / sqrt (?aMa * ?bMb))
 ⟨proof⟩

lemma *cosh-dist-perp-special-case*:
assumes $|x| < 1$ **and** $|y| < 1$
shows *cosh-dist* (*proj2-abs* (vector [x,0,1])) (*proj2-abs* (vector [0,y,1]))
 = (*cosh-dist* *K2-centre* (*proj2-abs* (vector [x,0,1])))
 * (*cosh-dist* *K2-centre* (*proj2-abs* (vector [0,y,1])))
(is *cosh-dist* ?pa ?pb = (*cosh-dist* ?po ?pa) * (*cosh-dist* ?po ?pb))
 ⟨proof⟩

lemma *K2-isometry-cross-ratio-endpoints-in-S*:
assumes a ∈ *hyp2* **and** b ∈ *hyp2* **and** *is-K2-isometry* J **and** $a \neq b$
shows *cross-ratio* (*apply-cltn2* (*endpoint-in-S* a b) J)
 (*apply-cltn2* (*endpoint-in-S* b a) J) (*apply-cltn2* a J) (*apply-cltn2* b J)
 = *cross-ratio* (*endpoint-in-S* a b) (*endpoint-in-S* b a) a b
(is *cross-ratio* ?pJ ?qJ ?aJ ?bJ = *cross-ratio* ?p ?q a b)
 ⟨proof⟩

lemma *K2-isometry-exp-2dist*:
assumes a ∈ *hyp2* **and** b ∈ *hyp2* **and** *is-K2-isometry* J
shows *exp-2dist* (*apply-cltn2* a J) (*apply-cltn2* b J) = *exp-2dist* a b
(is *exp-2dist* ?aJ ?bJ = -)
 ⟨proof⟩

lemma *K2-isometry-cosh-dist:*

assumes $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$ **and** *is-K2-isometry* J
shows $\text{cosh-dist } (\text{apply-cltn2 } a \ J) \ (\text{apply-cltn2 } b \ J) = \text{cosh-dist } a \ b$
{proof}

lemma *cosh-dist-perp:*

assumes $M\text{-perp } l \ m$ **and** $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$ **and** $c \in \text{hyp2}$
and *proj2-incident* $a \ l$ **and** *proj2-incident* $b \ l$
and *proj2-incident* $b \ m$ **and** *proj2-incident* $c \ m$
shows $\text{cosh-dist } a \ c = \text{cosh-dist } b \ a * \text{cosh-dist } b \ c$
{proof}

lemma *are-endpoints-in-S-ordered-cross-ratio:*

assumes *are-endpoints-in-S* $p \ q \ a \ b$
and $B_{\mathbb{R}} \ (\text{cart2-pt } a) \ (\text{cart2-pt } b) \ (\text{cart2-pt } p)$ (**is** $B_{\mathbb{R}}$ $?ca \ ?cb \ ?cp$)
shows $\text{cross-ratio } p \ q \ a \ b \geq 1$
{proof}

lemma *cross-ratio-S-S-hyp2-hyp2-positive:*

assumes *are-endpoints-in-S* $p \ q \ a \ b$
shows $\text{cross-ratio } p \ q \ a \ b > 0$
{proof}

lemma *cosh-dist-general:*

assumes *are-endpoints-in-S* $p \ q \ a \ b$
shows $\text{cosh-dist } a \ b$
 $= (\text{sqrt } (\text{cross-ratio } p \ q \ a \ b) + 1 / \text{sqrt } (\text{cross-ratio } p \ q \ a \ b)) / 2$
{proof}

lemma *exp-2dist-positive:*

assumes $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$
shows $\text{exp-2dist } a \ b > 0$
{proof}

lemma *cosh-dist-at-least-1:*

assumes $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$
shows $\text{cosh-dist } a \ b \geq 1$
{proof}

lemma *cosh-dist-positive:*

assumes $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$
shows $\text{cosh-dist } a \ b > 0$
{proof}

lemma *cosh-dist-perp-divide:*

assumes $M\text{-perp } l \ m$ **and** $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$ **and** $c \in \text{hyp2}$
and *proj2-incident* $a \ l$ **and** *proj2-incident* $b \ l$ **and** *proj2-incident* $b \ m$
and *proj2-incident* $c \ m$
shows $\text{cosh-dist } b \ c = \text{cosh-dist } a \ c / \text{cosh-dist } b \ a$

<proof>

lemma *real-hyp2-C-cross-ratio-endpoints-in-S:*

assumes $a \neq b$ **and** $a b \equiv_K c d$

shows $\text{cross-ratio } (\text{endpoint-in-S } (\text{Rep-hyp2 } a) (\text{Rep-hyp2 } b))$

$(\text{endpoint-in-S } (\text{Rep-hyp2 } b) (\text{Rep-hyp2 } a)) (\text{Rep-hyp2 } a) (\text{Rep-hyp2 } b)$

$= \text{cross-ratio } (\text{endpoint-in-S } (\text{Rep-hyp2 } c) (\text{Rep-hyp2 } d))$

$(\text{endpoint-in-S } (\text{Rep-hyp2 } d) (\text{Rep-hyp2 } c)) (\text{Rep-hyp2 } c) (\text{Rep-hyp2 } d)$

(is $\text{cross-ratio } ?p ?q ?a' ?b' = \text{cross-ratio } ?r ?s ?c' ?d'$ **)**

<proof>

lemma *real-hyp2-C-exp-2dist:*

assumes $a b \equiv_K c d$

shows $\text{exp-2dist } (\text{Rep-hyp2 } a) (\text{Rep-hyp2 } b)$

$= \text{exp-2dist } (\text{Rep-hyp2 } c) (\text{Rep-hyp2 } d)$

(is $\text{exp-2dist } ?a' ?b' = \text{exp-2dist } ?c' ?d'$ **)**

<proof>

lemma *real-hyp2-C-cosh-dist:*

assumes $a b \equiv_K c d$

shows $\text{cosh-dist } (\text{Rep-hyp2 } a) (\text{Rep-hyp2 } b)$

$= \text{cosh-dist } (\text{Rep-hyp2 } c) (\text{Rep-hyp2 } d)$

<proof>

lemma *cross-ratio-in-terms-of-cosh-dist:*

assumes *are-endpoints-in-S* $p q a b$

and $B_{\mathbb{R}} (\text{cart2-pt } a) (\text{cart2-pt } b) (\text{cart2-pt } p)$

shows $\text{cross-ratio } p q a b$

$= 2 * (\text{cosh-dist } a b)^2 + 2 * \text{cosh-dist } a b * \text{sqrt } ((\text{cosh-dist } a b)^2 - 1) - 1$

(is $?pqab = 2 * ?ab^2 + 2 * ?ab * \text{sqrt } (?ab^2 - 1) - 1$ **)**

<proof>

lemma *are-endpoints-in-S-cross-ratio-correct:*

assumes *are-endpoints-in-S* $p q a b$

shows *cross-ratio-correct* $p q a b$

<proof>

lemma *endpoints-in-S-cross-ratio-correct:*

assumes $a \neq b$ **and** $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$

shows *cross-ratio-correct* $(\text{endpoint-in-S } a b) (\text{endpoint-in-S } b a) a b$

<proof>

lemma *endpoints-in-S-perp-foot-cross-ratio-correct:*

assumes $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$ **and** $c \in \text{hyp2}$ **and** $a \neq b$

and *proj2-incident* $a l$ **and** *proj2-incident* $b l$

shows *cross-ratio-correct*

$(\text{endpoint-in-S } a b) (\text{endpoint-in-S } b a) a (\text{perp-foot } c l)$

(is *cross-ratio-correct* $?p ?q a ?d$ **)**

<proof>

lemma *cosh-dist-unique*:

assumes $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$ **and** $c \in \text{hyp2}$ **and** $p \in S$
and $B_{\mathbb{R}}(\text{cart2-pt } a) (\text{cart2-pt } b) (\text{cart2-pt } p)$ **(is** $B_{\mathbb{R}} ?ca ?cb ?cp)$
and $B_{\mathbb{R}}(\text{cart2-pt } a) (\text{cart2-pt } c) (\text{cart2-pt } p)$ **(is** $B_{\mathbb{R}} ?ca ?cc ?cp)$
and $\text{cosh-dist } a \ b = \text{cosh-dist } a \ c$ **(is** $?ab = ?ac)$
shows $b = c$

$\langle \text{proof} \rangle$

lemma *cosh-dist-swap*:

assumes $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$
shows $\text{cosh-dist } a \ b = \text{cosh-dist } b \ a$

$\langle \text{proof} \rangle$

lemma *exp-2dist-1-equal*:

assumes $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$ **and** $\text{exp-2dist } a \ b = 1$
shows $a = b$

$\langle \text{proof} \rangle$

8.11.1 A formula for a cross ratio involving a perpendicular foot

lemma *described-perp-foot-cross-ratio-formula*:

assumes $a \neq b$ **and** $c \in \text{hyp2}$ **and** *are-endpoints-in-S* $p \ q \ a \ b$
and *proj2-incident* $p \ l$ **and** *proj2-incident* $q \ l$ **and** *M-perp* $l \ m$
and *proj2-incident* $d \ l$ **and** *proj2-incident* $d \ m$ **and** *proj2-incident* $c \ m$
shows $\text{cross-ratio } p \ q \ d \ a$
 $= (\text{cosh-dist } b \ c * \text{sqrt}(\text{cross-ratio } p \ q \ a \ b) - \text{cosh-dist } a \ c)$
 $/ (\text{cosh-dist } a \ c * \text{cross-ratio } p \ q \ a \ b$
 $- \text{cosh-dist } b \ c * \text{sqrt}(\text{cross-ratio } p \ q \ a \ b))$
(is $?pqda = (?bc * \text{sqrt } ?pqab - ?ac) / (?ac * ?pqab - ?bc * \text{sqrt } ?pqab)$
 $\langle \text{proof} \rangle$

lemma *perp-foot-cross-ratio-formula*:

assumes $a \in \text{hyp2}$ **and** $b \in \text{hyp2}$ **and** $c \in \text{hyp2}$ **and** $a \neq b$
shows $\text{cross-ratio}(\text{endpoint-in-S } a \ b) (\text{endpoint-in-S } b \ a)$
 $(\text{perp-foot } c (\text{proj2-line-through } a \ b)) \ a$
 $= (\text{cosh-dist } b \ c * \text{sqrt}(\text{exp-2dist } a \ b) - \text{cosh-dist } a \ c)$
 $/ (\text{cosh-dist } a \ c * \text{exp-2dist } a \ b - \text{cosh-dist } b \ c * \text{sqrt}(\text{exp-2dist } a \ b))$
(is $\text{cross-ratio } ?p \ ?q \ ?d \ a$
 $= (?bc * \text{sqrt } ?pqab - ?ac) / (?ac * ?pqab - ?bc * \text{sqrt } ?pqab)$

$\langle \text{proof} \rangle$

8.12 The Klein–Beltrami model satisfies axiom 5

lemma *statement69*:

assumes $a \ b \equiv_K a' \ b'$ **and** $b \ c \equiv_K b' \ c'$ **and** $a \ c \equiv_K a' \ c'$
shows $\exists J. \text{is-K2-isometry } J$
 $\wedge \text{hyp2-cltn2 } a \ J = a' \wedge \text{hyp2-cltn2 } b \ J = b' \wedge \text{hyp2-cltn2 } c \ J = c'$

$\langle \text{proof} \rangle$

theorem *hyp2-axiom5*:

$$\begin{aligned} & \forall a b c d a' b' c' d'. \\ & a \neq b \wedge B_K a b c \wedge B_K a' b' c' \wedge a b \equiv_K a' b' \wedge b c \equiv_K b' c' \\ & \wedge a d \equiv_K a' d' \wedge b d \equiv_K b' d' \\ & \longrightarrow c d \equiv_K c' d' \end{aligned}$$

<proof>

interpretation *hyp2: tarski-first5 real-hyp2-C real-hyp2-B*

<proof>

8.13 The Klein–Beltrami model satisfies axioms 6, 7, and 11

theorem *hyp2-axiom6*: $\forall a b. B_K a b a \longrightarrow a = b$

<proof>

lemma *between-inverse*:

assumes B_R (*hyp2-rep* p) v (*hyp2-rep* q)

shows *hyp2-rep* (*hyp2-abs* v) = v

<proof>

lemma *between-switch*:

assumes B_R (*hyp2-rep* p) v (*hyp2-rep* q)

shows $B_K p$ (*hyp2-abs* v) q

<proof>

theorem *hyp2-axiom7*:

$$\forall a b c p q. B_K a p c \wedge B_K b q c \longrightarrow (\exists x. B_K p x b \wedge B_K q x a)$$

<proof>

theorem *hyp2-axiom11*:

$$\forall X Y. (\exists a. \forall x y. x \in X \wedge y \in Y \longrightarrow B_K a x y)$$

$$\longrightarrow (\exists b. \forall x y. x \in X \wedge y \in Y \longrightarrow B_K x b y)$$

<proof>

interpretation *tarski-absolute-space real-hyp2-C real-hyp2-B*

<proof>

8.14 The Klein–Beltrami model satisfies the dimension-specific axioms

lemma *hyp2-rep-abs-examples*:

shows *hyp2-rep* (*hyp2-abs* 0) = 0 (**is** *hyp2-rep* ?a = ?ca)

and *hyp2-rep* (*hyp2-abs* (*vector* [1/2,0])) = *vector* [1/2,0]

(**is** *hyp2-rep* ?b = ?cb)

and *hyp2-rep* (*hyp2-abs* (*vector* [0,1/2])) = *vector* [0,1/2]

(**is** *hyp2-rep* ?c = ?cc)

and *hyp2-rep* (*hyp2-abs* (*vector* [1/4,1/4])) = *vector* [1/4,1/4]

(**is** *hyp2-rep* ?d = ?cd)

and *hyp2-rep* (*hyp2-abs* (*vector* [1/2,1/2])) = *vector* [1/2,1/2]

(is hyp2-rep ?t = ?ct)
<proof>

theorem hyp2-axiom8: $\exists a b c. \neg B_K a b c \wedge \neg B_K b c a \wedge \neg B_K c a b$
<proof>

theorem hyp2-axiom9:
 $\forall p q a b c. p \neq q \wedge a p \equiv_K a q \wedge b p \equiv_K b q \wedge c p \equiv_K c q$
 $\longrightarrow B_K a b c \vee B_K b c a \vee B_K c a b$
<proof>

interpretation hyp2: tarski-absolute real-hyp2-C real-hyp2-B
<proof>

8.15 The Klein–Beltrami model violates the Euclidean axiom

theorem hyp2-axiom10-false:
shows $\neg (\forall a b c d t. B_K a d t \wedge B_K b d c \wedge a \neq d$
 $\longrightarrow (\exists x y. B_K a b x \wedge B_K a c y \wedge B_K x t y))$
<proof>

theorem hyp2-not-tarski: $\neg (\text{tarski real-hyp2-C real-hyp2-B})$
<proof>

Therefore axiom 10 is independent.

end

References

- [1] K. Borsuk and W. Szmielew. *Foundations of Geometry: Euclidean and Bolyai-Lobachevskian Geometry; Projective Geometry*. North-Holland Publishing Company, 1960. Translated from Polish by Erwin Marquit.
- [2] T. J. M. Makarios. A mechanical verification of the independence of Tarski’s Euclidean axiom. Master’s thesis, Victoria University of Wellington, New Zealand, 2012. <http://researcharchive.vuw.ac.nz/handle/10063/2315>.
- [3] W. Schwabhäuser, W. Szmielew, and A. Tarski. *Metamathematische Methoden in der Geometrie*. Springer-Verlag, 1983.