Poincaré Disc Model

Danijela Simić Filip Marić Pierre Boutry

April 9, 2025

Abstract

We describe formalization of the Poincaré disc model of hyperbolic geometry within the Isabelle/HOL proof assistant. The model is defined within the extended complex plane (one dimensional complex projective space $\mathbb{C}P^1$), formalized in the AFP entry "Complex Geometry" [6]. Points, lines, congruence of pairs of points, betweenness of triples of points, circles, and isometries are defined within the model. It is shown that the model satisfies all Tarski's axioms except the Euclid's axiom. It is shown that it satisfies its negation and the limiting parallels axiom (which proves it to be a model of hyperbolic geometry).

Contents

1 Introduction

Poincaré disc is a model of hyperbolic geometry. That fact has been a mathematical folklore for more than 100 years. However, up to the best of our knowledge, fully precise, formal proofs of this fact are lacking. In this paper we present a formalization of the Poincaré disc model in Isabelle/HOL, introduce its basic notions (h-points, h-lines, h-congruence, h-isometries, h-betweenness) and prove that it models Tarski's axioms except for Euclid's axiom. We shown that is satisfies the negation of Euclid's axiom, and, moreover, the existence of limiting parallels axiom. The model is defined within the extended complex plane, which has been described quite precisely by Schwerdfeger [8] and formalized in the previous work of the first two authors [5].

Related work. In 1840 Lobachevsky [3] published developments about non-Euclidean geometry. Hyperbolic geometry is studied through many of its models. The concept of a projective disc model was introduced by Klein while Poincaré investigated the half-plane model proposed by Liouville and Beltrami and primarily studied the isometries of the hyperbolic plane that preserve orientation. In this paper, we focus on the formalization of the latter.

Regarding non-Euclidean geometry, Makarios showed the independence of Euclid's axiom [4]. He did so by formalizing that the Klein–Beltrami model is a model of Tarski's axioms at the exception of Euclid's axiom. Latter Coghetto formalized the Klein-Beltrami model within Mizar [1, 2].

2 Background theories

2.1 Hyperbolic Functions

In this section hyperbolic cosine and hyperbolic sine functions are introduced and some of their properties needed for further development are proved.

theory *Hyperbolic-Functions*
  imports *Complex-Main Complex-Geometry.More-Complex*
begin

lemma *arcosh-eq-iff*:
  fixes *x y::real*
  assumes $x \geq 1$ $y \geq 1$
  shows $arcosh\ x = arcosh\ y \longleftrightarrow x = y$
  ⟨*proof*⟩

lemma *cosh-gt-1* [*simp*]:
  fixes *x ::real*
  assumes $x > 0$
  shows $cosh\ x > 1$
  ⟨*proof*⟩

lemma *cosh-eq-iff*:
  fixes *x y::real*
  assumes $x \geq 0$ $y \geq 0$
  shows $cosh\ x = cosh\ y \longleftrightarrow x = y$
  ⟨*proof*⟩

lemma *arcosh-mono*:
  fixes *x y::real*
  assumes $x \geq 1$ $y \geq 1$
  shows $arcosh\ x \geq arcosh\ y \longleftrightarrow x \geq y$
  ⟨*proof*⟩

lemma *arcosh-add*:
  fixes *x y::real*
  assumes $x \geq 1$ $y \geq 1$
  shows $arcosh\ x + arcosh\ y = arcosh\ (x*y + sqrt((x^2 - 1)*(y^2 - 1)))$
⟨*proof*⟩

lemma *arcosh-double*:
  fixes x :: *real*
  assumes $x \geq 1$
  shows $2 * arcosh\ x = arcosh\ (2{*}x^2 - 1)$
  ⟨*proof*⟩

end

3 Tarski axioms

In this section we introduce axioms of Tarski [7] trough a series of locales.

theory *Tarski*
imports *Main*
begin

The first locale assumes all Tarski axioms except for the Euclid's axiom and the continuity axiom and corresponds to absolute geometry.

locale *TarskiAbsolute* =
  fixes *cong* :: $'p \Rightarrow 'p \Rightarrow 'p \Rightarrow 'p \Rightarrow bool$
  fixes *betw* :: $'p \Rightarrow 'p \Rightarrow 'p \Rightarrow bool$
  assumes *cong-reflexive*: *cong x y y x*
  assumes *cong-transitive*: *cong x y z u* ∧ *cong x y v w* ⟶ *cong z u v w*
  assumes *cong-identity*: *cong x y z z* ⟶ *x* = *y*
  assumes *segment-construction*: ∃ *z*. *betw x y z* ∧ *cong y z a b*
  assumes *five-segment*: *x* ≠ *y* ∧ *betw x y z* ∧ *betw x′ y′ z′* ∧ *cong x y x′ y′* ∧ *cong y z y′ z′* ∧ *cong x u x′ u′* ∧ *cong y u y′ u′* ⟶ *cong z u z′ u′*
  assumes *betw-identity*: *betw x y x* ⟶ *x* = *y*
  assumes *Pasch*: *betw x u z* ∧ *betw y v z* ⟶ (∃ *a*. *betw u a y* ∧ *betw x a v*)
  assumes *lower-dimension*: ∃ *a*. ∃ *b*. ∃ *c*. ¬ *betw a b c* ∧ ¬ *betw b c a* ∧ ¬ *betw c a b*
  assumes *upper-dimension*: *cong x u x v* ∧ *cong y u y v* ∧ *cong z u z v* ∧ *u* ≠ *v* ⟶ *betw x y z* ∨ *betw y z x* ∨ *betw z x y*
begin

The following definitions are used to specify axioms in the following locales.

Point p is on line ab.

definition *on-line* **where**
  on-line p a b ⟷ *betw p a b* ∨ *betw a p b* ∨ *betw a b p*

Point p is on ray ab.

definition *on-ray* **where**
  on-ray p a b ⟷ *betw a p b* ∨ *betw a b p*

Point p is inside angle abc.

definition *in-angle* **where**
  in-angle p a b c ⟷ *b* ≠ *a* ∧ *b* ≠ *c* ∧ *p* ≠ *b* ∧ (∃ *x*. *betw a x c* ∧ *x* ≠ *a* ∧ *x* ≠ *c* ∧ *on-ray p b x*)

Ray $r_a r_b$ meets the line $l_a l_b$.

definition *ray-meets-line* **where**
  ray-meets-line ra rb la lb ⟷ (∃ *x*. *on-ray x ra rb* ∧ *on-line x la lb*)

end

The second locales adds the negation of Euclid's axiom and limiting parallels and corresponds to hyperbolic geometry.

locale *TarskiHyperbolic* = *TarskiAbsolute* +
  assumes *euclid-negation*: ∃ *a b c d t*. *betw a d t* ∧ *betw b d c* ∧ *a* ≠ *d* ∧ (∀ *x y*. *betw a b x* ∧ *betw a c y* ⟶ ¬ *betw x t y*)
  assumes *limiting-parallels*: ¬ *on-line a x1 x2* ⟹
    (∃ *a1 a2*. ¬ *on-line a a1 a2* ∧
        ¬ *ray-meets-line a a1 x1 x2* ∧
        ¬ *ray-meets-line a a2 x1 x2* ∧
        (∀ *a′*. *in-angle a′ a1 a a2* ⟶ *ray-meets-line a a′ x1 x2*))

The third locale adds the continuity axiom and corresponds to elementary hyperbolic geometry.

locale *ElementaryTarskiHyperbolic* = *TarskiHyperbolic* +
  assumes *continuity*: $[\![\exists\ a.\ \forall\ x.\ \forall\ y.\ \varphi\ x \wedge \psi\ y \longrightarrow betw\ a\ x\ y]\!] \Longrightarrow \exists\ b.\ \forall\ x.\ \forall\ y.\ \varphi\ x \wedge \psi\ y \longrightarrow betw\ x\ b\ y$

end

4 H-lines in the Poincaré model

theory *Poincare-Lines*
  imports *Complex-Geometry.Unit-Circle-Preserving-Moebius Complex-Geometry.Circlines-Angle*
begin

4.1 Definition and basic properties of h-lines

H-lines in the Poincaré model are either line segments passing trough the origin or segments (within the unit disc) of circles that are perpendicular to the unit circle. Algebraically these are circlines that are represented by Hermitean matrices of the form

$$H = \begin{pmatrix} A & B \\ \overline{B} & A \end{pmatrix},$$

for $A \in \mathbb{R}$, and $B \in \mathbb{C}$, and $|B|^2 > A^2$, where the circline equation is the usual one: $z^*Hz = 0$, for homogenous coordinates z.

definition *is-poincare-line-cmat* :: *complex-mat* ⇒ *bool* **where**
  [*simp*]: *is-poincare-line-cmat H* ⟷
        (*let* (*A*, *B*, *C*, *D*) = *H*
          in hermitean (*A*, *B*, *C*, *D*) ∧ *A* = *D* ∧ (*cmod B*)^2 > (*cmod A*)^2)

lift-definition *is-poincare-line-clmat* :: *circline-mat* ⇒ *bool* **is** *is-poincare-line-cmat*
  ⟨*proof*⟩

We introduce the predicate that checks if a given complex matrix is a matrix of a h-line in the Poincaré model, and then by means of the lifting package lift it to the type of non-zero Hermitean matrices, and then to circlines (that are equivalence classes of such matrices).

lift-definition *is-poincare-line* :: *circline* ⇒ *bool* **is** *is-poincare-line-clmat*
⟨*proof*⟩

lemma *is-poincare-line-mk-circline*:
  assumes (*A*, *B*, *C*, *D*) ∈ *hermitean-nonzero*
  shows *is-poincare-line* (*mk-circline A B C D*) ⟷ (*cmod B*)^2 > (*cmod A*)^2 ∧ *A* = *D*
  ⟨*proof*⟩

Abstract characterisation of *is-poincare-line* predicate: H-lines in the Poincaré model are real circlines (circlines with the negative determinant) perpendicular to the unit circle.

lemma *is-poincare-line-iff*:
  shows *is-poincare-line H* ⟷ *circline-type H* = *−1* ∧ *perpendicular H unit-circle*
  ⟨*proof*⟩

The *x-axis* is an h-line.

lemma *is-poincare-line-x-axis* [*simp*]:
  shows *is-poincare-line x-axis*
  ⟨*proof*⟩

The *unit-circle* is not an h-line.

lemma *not-is-poincare-line-unit-circle* [*simp*]:
  shows ¬ *is-poincare-line unit-circle*
  ⟨*proof*⟩

4.1.1 Collinear points

Points are collinear if they all belong to an h-line.

definition *poincare-collinear* :: *complex-homo set* ⇒ *bool* **where**
  poincare-collinear S ⟷ (∃ *p*. *is-poincare-line p* ∧ *S* ⊆ *circline-set p*)

4.1.2 H-lines and inversion

Every h-line in the Poincaré model contains the inverse (wrt. the unit circle) of each of its points (note that at most one of them belongs to the unit disc).

lemma *is-poincare-line-inverse-point*:
  assumes *is-poincare-line H u* ∈ *circline-set H*
  shows *inversion u* ∈ *circline-set H*
  ⟨*proof*⟩

Every h-line in the Poincaré model and is invariant under unit circle inversion.

lemma *circline-inversion-poincare-line*:
  assumes *is-poincare-line H*
  shows *circline-inversion H* = *H*
⟨*proof*⟩

4.1.3 Classification of h-lines into Euclidean segments and circles

If an h-line contains zero, than it also contains infinity (the inverse point of zero) and is by definition an Euclidean line.

lemma *is-poincare-line-trough-zero-trough-infty* [*simp*]:
  assumes *is-poincare-line l* **and** 0_h ∈ *circline-set l*
  shows ∞_h ∈ *circline-set l*
  ⟨*proof*⟩

lemma *is-poincare-line-trough-zero-is-line*:
  assumes *is-poincare-line l* **and** 0_h ∈ *circline-set l*
  shows *is-line l*
  ⟨*proof*⟩

If an h-line does not contain zero, than it also does not contain infinity (the inverse point of zero) and is by definition an Euclidean circle.

lemma *is-poincare-line-not-trough-zero-not-trough-infty* [*simp*]:
  assumes *is-poincare-line l*
  assumes 0_h ∉ *circline-set l*
  shows ∞_h ∉ *circline-set l*
  ⟨*proof*⟩

lemma *is-poincare-line-not-trough-zero-is-circle*:
  assumes *is-poincare-line l* 0_h ∉ *circline-set l*
  shows *is-circle l*
  ⟨*proof*⟩

4.1.4 Points on h-line

Each h-line in the Poincaré model contains at least two different points within the unit disc.

First we prove an auxiliary lemma.

lemma *ex-is-poincare-line-points'*:
  assumes *i12*: *i1* ∈ *circline-set H* ∩ *unit-circle-set*
    i2 ∈ *circline-set H* ∩ *unit-circle-set*
    i1 ≠ *i2*
  assumes *a*: *a* ∈ *circline-set H a* ∉ *unit-circle-set*
  shows ∃ *b*. *b* ≠ *i1* ∧ *b* ≠ *i2* ∧ *b* ≠ *a* ∧ *b* ≠ *inversion a* ∧ *b* ∈ *circline-set H*
⟨*proof*⟩

Now we can prove the statement.

lemma *ex-is-poincare-line-points*:
  assumes *is-poincare-line H*
  shows ∃ *u v*. *u* ∈ *unit-disc* ∧ *v* ∈ *unit-disc* ∧ *u* ≠ *v* ∧ {*u*, *v*} ⊆ *circline-set H*
⟨*proof*⟩

4.1.5 H-line uniqueness

There is no more than one h-line that contains two different h-points (in the disc).

lemma *unique-is-poincare-line*:
  assumes *in-disc*: $u \in$ *unit-disc* $v \in$ *unit-disc* $u \neq v$
  assumes *pl*: *is-poincare-line l1 is-poincare-line l2*
  assumes *on-l*: $\{u, v\} \subseteq$ *circline-set l1* $\cap$ *circline-set l2*
  shows $l1 = l2$
⟨*proof*⟩

For the rest of our formalization it is often useful to consider points on h-lines that are not within the unit disc. Many lemmas in the rest of this section will have such generalizations.

There is no more than one h-line that contains two different and not mutually inverse points (not necessary in the unit disc).

lemma *unique-is-poincare-line-general*:
  assumes *different*: $u \neq v$ $u \neq$ *inversion* v
  assumes *pl*: *is-poincare-line l1 is-poincare-line l2*
  assumes *on-l*: $\{u, v\} \subseteq$ *circline-set l1* $\cap$ *circline-set l2*
  shows $l1 = l2$
⟨*proof*⟩

The only h-line that goes trough zero and a non-zero point on the x-axis is the x-axis.

lemma *is-poincare-line-0-real-is-x-axis*:
  assumes *is-poincare-line l* $0_h \in$ *circline-set l*
    $x \in$ *circline-set l* $\cap$ *circline-set x-axis* $x \neq 0_h$ $x \neq \infty_h$
  shows $l =$ *x-axis*
  ⟨*proof*⟩

The only h-line that goes trough zero and a non-zero point on the y-axis is the y-axis.

lemma *is-poincare-line-0-imag-is-y-axis*:
  assumes *is-poincare-line l* $0_h \in$ *circline-set l*
    $y \in$ *circline-set l* $\cap$ *circline-set y-axis* $y \neq 0_h$ $y \neq \infty_h$
  shows $l =$ *y-axis*
  ⟨*proof*⟩

4.1.6 H-isometries preserve h-lines

H-isometries are defined as homographies (actions of Möbius transformations) and antihomographies (compositions of actions of Möbius transformations with conjugation) that fix the unit disc (map it onto itself). They also map h-lines onto h-lines

We prove a bit more general lemma that states that all Möbius transformations that fix the unit circle (not necessary the unit disc) map h-lines onto h-lines

lemma *unit-circle-fix-preserve-is-poincare-line* [*simp*]:
  assumes *unit-circle-fix M is-poincare-line H*
  shows *is-poincare-line* (*moebius-circline M H*)
  ⟨*proof*⟩

lemma *unit-circle-fix-preserve-is-poincare-line-iff* [*simp*]:
  assumes *unit-circle-fix M*
  shows *is-poincare-line* (*moebius-circline M H*) $\longleftrightarrow$ *is-poincare-line H*
  ⟨*proof*⟩

Since h-lines are preserved by transformations that fix the unit circle, so is collinearity.

lemma *unit-disc-fix-preserve-poincare-collinear* [*simp*]:
  assumes *unit-circle-fix M poincare-collinear A*
  shows *poincare-collinear* (*moebius-pt M ‘ A*)
  ⟨*proof*⟩

lemma *unit-disc-fix-preserve-poincare-collinear-iff* [*simp*]:
  assumes *unit-circle-fix M*
  shows *poincare-collinear* (*moebius-pt M ‘ A*) $\longleftrightarrow$ *poincare-collinear A*
  ⟨*proof*⟩

lemma *unit-disc-fix-preserve-poincare-collinear3* [*simp*]:
  assumes *unit-disc-fix M*
  shows *poincare-collinear* {*moebius-pt M u*, *moebius-pt M v*, *moebius-pt M w*} $\longleftrightarrow$
    poincare-collinear {*u*, *v*, *w*}
  ⟨*proof*⟩

Conjugation is also an h-isometry and it preserves h-lines.

lemma *is-poincare-line-conjugate-circline* [*simp*]:
  assumes *is-poincare-line H*
  shows *is-poincare-line* (*conjugate-circline H*)
  ⟨*proof*⟩

lemma *is-poincare-line-conjugate-circline-iff* [*simp*]:
  shows *is-poincare-line* (*conjugate-circline H*) $\longleftrightarrow$ *is-poincare-line H*
  ⟨*proof*⟩

Since h-lines are preserved by conjugation, so is collinearity.

lemma *conjugate-preserve-poincare-collinear* [*simp*]:
  assumes *poincare-collinear A*
  shows *poincare-collinear* (*conjugate* ‘ *A*)
  ⟨*proof*⟩

lemma *conjugate-conjugate* [*simp*]: *conjugate* ‘ *conjugate* ‘ *A* = *A*
  ⟨*proof*⟩

lemma *conjugate-preserve-poincare-collinear-iff* [*simp*]:
  shows *poincare-collinear* (*conjugate* ‘ *A*) $\longleftrightarrow$ *poincare-collinear A*
  ⟨*proof*⟩

4.1.7 Mapping h-lines to x-axis

Each h-line in the Poincaré model can be mapped onto the x-axis (by a unit-disc preserving Möbius transformation).

lemma *ex-unit-disc-fix-is-poincare-line-to-x-axis*:
  assumes *is-poincare-line l*
  shows ∃ *M*. *unit-disc-fix M* ∧ *moebius-circline M l* = *x-axis*
⟨*proof*⟩

When proving facts about h-lines, without loss of generality it can be assumed that h-line is the x-axis (if the property being proved is invariant under Möbius transformations that fix the unit disc).

lemma *wlog-line-x-axis*:
  assumes *is-line*: *is-poincare-line H*
  assumes *x-axis*: *P x-axis*
  assumes *preserves*: $\bigwedge$ *M*. ⟦*unit-disc-fix M*; *P* (*moebius-circline M H*)⟧ $\Longrightarrow$ *P H*
  shows *P H*
  ⟨*proof*⟩

4.2 Construction of the h-line between the two given points

Next we show how to construct the (unique) h-line between the two given points in the Poincaré model

Geometrically, h-line can be constructed by finding the inverse point of one of the two points and by constructing the circle (or line) trough it and the two given points.
Algebraically, for two given points u and v in $\mathbb{C}$, the h-line matrix coefficients can be $A = i \cdot (u\overline{v} - v\overline{u})$ and $B = i \cdot (v(|u|^2 + 1) - u(|v|^2 + 1))$.
We need to extend this to homogenous coordinates. There are several degenerate cases.
- If $\{z, w\} = \{0_h, \infty_h\}$ then there is no unique h-line (any line trough zero is an h-line).
- If z and w are mutually inverse, then the construction fails (both geometric and algebraic).
- If z and w are different points on the unit circle, then the standard construction fails (only geometric).
- None of this problematic cases occur when z and w are inside the unit disc.

We express the construction algebraically, and construct the Hermitean circline matrix for the two points given in homogenous coordinates. It works correctly in all cases except when the two points are the same or are mutually inverse.

definition *mk-poincare-line-cmat* :: *real* ⇒ *complex* ⇒ *complex-mat* **where**
  [*simp*]: *mk-poincare-line-cmat A B* = (*cor A*, *B*, *cnj B*, *cor A*)

lemma *mk-poincare-line-cmat-zero-iff*:
  mk-poincare-line-cmat A B = *mat-zero* ⟷ *A* = *0* ∧ *B* = *0*
  ⟨*proof*⟩

lemma *mk-poincare-line-cmat-hermitean*
  [*simp*]: *hermitean* (*mk-poincare-line-cmat A B*)
  ⟨*proof*⟩

lemma *mk-poincare-line-cmat-scale*:
  cor k $*_{sm}$ *mk-poincare-line-cmat A B* = *mk-poincare-line-cmat* (*k* ∗ *A*) (*k* ∗ *B*)
  ⟨*proof*⟩

definition *poincare-line-cvec-cmat* :: *complex-vec* ⇒ *complex-vec* ⇒ *complex-mat* **where**
  [*simp*]: *poincare-line-cvec-cmat z w* =
      (*let* (*z1*, *z2*) = *z*;
          (*w1*, *w2*) = *w*;
          nom = *w1*∗*cnj w2*∗(*z1*∗*cnj z1* + *z2*∗*cnj z2*) − *z1*∗*cnj z2*∗(*w1*∗*cnj w1* + *w2*∗*cnj w2*);
          den = *z1*∗*cnj z2*∗*cnj w1*∗*w2* − *w1*∗*cnj w2*∗*cnj z1*∗*z2*
       in if den ≠ *0 then*
            mk-poincare-line-cmat (*Re*(i∗*den*)) (i∗*nom*)
          else if z1∗*cnj z2* ≠ *0 then*
            mk-poincare-line-cmat 0 (i∗*z1*∗*cnj z2*)
          else if w1∗*cnj w2* ≠ *0 then*
            mk-poincare-line-cmat 0 (i∗*w1*∗*cnj w2*)
          else
            mk-poincare-line-cmat 0 i)

lemma *poincare-line-cvec-cmat-AeqD*:
  assumes *poincare-line-cvec-cmat z w* = (*A*, *B*, *C*, *D*)
  shows *A* = *D*
  ⟨*proof*⟩

lemma *poincare-line-cvec-cmat-hermitean* [*simp*]:
  shows *hermitean* (*poincare-line-cvec-cmat z w*)
  ⟨*proof*⟩

lemma *poincare-line-cvec-cmat-nonzero* [*simp*]:
  assumes *z* ≠ *vec-zero w* ≠ *vec-zero*
  shows *poincare-line-cvec-cmat z w* ≠ *mat-zero*
⟨*proof*⟩

lift-definition *poincare-line-hcoords-clmat* :: *complex-homo-coords* ⇒ *complex-homo-coords* ⇒ *circline-mat* **is** *poincare-line-cvec-cmat*
  ⟨*proof*⟩

lift-definition *poincare-line* :: *complex-homo* ⇒ *complex-homo* ⇒ *circline* **is** *poincare-line-hcoords-clmat*
⟨*proof*⟩

4.2.1 Correctness of the construction

For finite points, our definition matches the classic algebraic definition for points in ℂ (given in ordinary, not homogenous coordinates).

lemma *poincare-line-non-homogenous*:
  assumes *u* ≠ ∞_h *v* ≠ ∞_h *u* ≠ *v u* ≠ *inversion v*
  shows *let u'* = *to-complex u*; *v'* = *to-complex v*;
        A = i ∗ (*u'* ∗ *cnj v'* − *v'* ∗ *cnj u'*);
        B = i ∗ (*v'* ∗ ((*cmod u'*)^2 + *1*) − *u'* ∗ ((*cmod v'*)^2 + *1*))
      in poincare-line u v = *mk-circline A B* (*cnj B*) *A*
  ⟨*proof*⟩

Our construction (in homogenous coordinates) always yields an h-line that contain two starting points (this also holds for all degenerate cases except when points are the same).

lemma *poincare-line* [*simp*]:
assumes $z \neq w$
shows *on-circline* (*poincare-line z w*) *z*
on-circline (*poincare-line z w*) *w*
⟨*proof*⟩

lemma *poincare-line-circline-set* [*simp*]:
assumes $z \neq w$
shows $z \in$ *circline-set* (*poincare-line z w*)
$w \in$ *circline-set* (*poincare-line z w*)
⟨*proof*⟩

When the points are different, the constructed line matrix always has a negative determinant

lemma *poincare-line-type*:
assumes $z \neq w$
shows *circline-type* (*poincare-line z w*) = −*1*
⟨*proof*⟩

The constructed line is an h-line in the Poincaré model (in all cases when the two points are different)

lemma *is-poincare-line-poincare-line* [*simp*]:
assumes $z \neq w$
shows *is-poincare-line* (*poincare-line z w*)
⟨*proof*⟩

When the points are different, the constructed h-line between two points also contains their inverses

lemma *poincare-line-inversion*:
assumes $z \neq w$
shows *on-circline* (*poincare-line z w*) (*inversion z*)
on-circline (*poincare-line z w*) (*inversion w*)
⟨*proof*⟩

When the points are different, the onstructed h-line between two points contains the inverse of its every point

lemma *poincare-line-inversion-full*:
assumes $u \neq v$
assumes *on-circline* (*poincare-line u v*) *x*
shows *on-circline* (*poincare-line u v*) (*inversion x*)
⟨*proof*⟩

4.2.2 Existence of h-lines

There is an h-line trough every point in the Poincaré model

lemma *ex-poincare-line-one-point*:
shows ∃ *l*. *is-poincare-line l* ∧ $z \in$ *circline-set l*
⟨*proof*⟩

lemma *poincare-collinear-singleton* [*simp*]:
assumes $u \in$ *unit-disc*
shows *poincare-collinear* {*u*}
⟨*proof*⟩

There is an h-line trough every two points in the Poincaré model

lemma *ex-poincare-line-two-points*:
assumes $z \neq w$
shows ∃ *l*. *is-poincare-line l* ∧ $z \in$ *circline-set l* ∧ $w \in$ *circline-set l*
⟨*proof*⟩

lemma *poincare-collinear-doubleton* [*simp*]:
assumes $u \in$ *unit-disc* $v \in$ *unit-disc*
shows *poincare-collinear* {*u*, *v*}
⟨*proof*⟩

4.2.3 Uniqueness of h-lines

The only h-line between two points is the one obtained by the line-construction.

First we show this only for two different points inside the disc.

lemma *unique-poincare-line*:
  assumes *in-disc*: $u \neq v$ $u \in$ *unit-disc* $v \in$ *unit-disc*
  assumes *on-l*: $u \in$ *circline-set l* $v \in$ *circline-set l is-poincare-line l*
  shows $l =$ *poincare-line u v*
  ⟨*proof*⟩

The assumption that the points are inside the disc can be relaxed.

lemma *unique-poincare-line-general*:
  assumes *in-disc*: $u \neq v$ $u \neq$ *inversion v*
  assumes *on-l*: $u \in$ *circline-set l* $v \in$ *circline-set l is-poincare-line l*
  shows $l =$ *poincare-line u v*
  ⟨*proof*⟩

The explicit line construction enables us to prove that there exists a unique h-line through any given two h-points (uniqueness part was already shown earlier).

First we show this only for two different points inside the disc.

lemma *ex1-poincare-line*:
  assumes $u \neq v$ $u \in$ *unit-disc* $v \in$ *unit-disc*
  shows $\exists!$ *l*. *is-poincare-line l* $\wedge$ $u \in$ *circline-set l* $\wedge$ $v \in$ *circline-set l*
⟨*proof*⟩

The assumption that the points are in the disc can be relaxed.

lemma *ex1-poincare-line-general*:
  assumes $u \neq v$ $u \neq$ *inversion v*
  shows $\exists!$ *l*. *is-poincare-line l* $\wedge$ $u \in$ *circline-set l* $\wedge$ $v \in$ *circline-set l*
⟨*proof*⟩

4.2.4 Some consequences of line uniqueness

H-line uv is the same as the h-line vu.

lemma *poincare-line-sym*:
  assumes $u \in$ *unit-disc* $v \in$ *unit-disc* $u \neq v$
  shows *poincare-line u v* = *poincare-line v u*
  ⟨*proof*⟩

lemma *poincare-line-sym-general*:
  assumes $u \neq v$ $u \neq$ *inversion v*
  shows *poincare-line u v* = *poincare-line v u*
  ⟨*proof*⟩

Each h-line is the h-line constructed out of its two arbitrary different points.

lemma *ex-poincare-line-points*:
  assumes *is-poincare-line H*
  shows $\exists$ *u v*. $u \in$ *unit-disc* $\wedge$ $v \in$ *unit-disc* $\wedge$ $u \neq v$ $\wedge$ H = *poincare-line u v*
  ⟨*proof*⟩

If an h-line contains two different points on x-axis/y-axis then it is the x-axis/y-axis.

lemma *poincare-line-0-real-is-x-axis*:
  assumes $x \in$ *circline-set x-axis* $x \neq 0_h$ $x \neq \infty_h$
  shows *poincare-line* 0_h x = *x-axis*
  ⟨*proof*⟩

lemma *poincare-line-0-imag-is-y-axis*:
  assumes $y \in$ *circline-set y-axis* $y \neq 0_h$ $y \neq \infty_h$
  shows *poincare-line* 0_h y = *y-axis*
  ⟨*proof*⟩

lemma *poincare-line-x-axis*:

assumes $x \in$ *unit-disc* $y \in$ *unit-disc* $x \in$ *circline-set x-axis* $y \in$ *circline-set x-axis* $x \neq y$
shows *poincare-line* x y = *x-axis*
⟨*proof*⟩

lemma *poincare-line-minus-one-one* [*simp*]:
shows *poincare-line* (*of-complex* (-1)) (*of-complex* 1) = *x-axis*
⟨*proof*⟩

4.2.5 Transformations of constructed lines

Unit dics preserving Möbius transformations preserve the h-line construction

lemma *unit-disc-fix-preserve-poincare-line* [*simp*]:
assumes *unit-disc-fix M u* $\in$ *unit-disc* $v \in$ *unit-disc* $u \neq v$
shows *poincare-line* (*moebius-pt M u*) (*moebius-pt M v*) = *moebius-circline M* (*poincare-line u v*)
⟨*proof*⟩

Conjugate preserve the h-line construction

lemma *conjugate-preserve-poincare-line* [*simp*]:
assumes $u \in$ *unit-disc* $v \in$ *unit-disc* $u \neq v$
shows *poincare-line* (*conjugate u*) (*conjugate v*) = *conjugate-circline* (*poincare-line u v*)
⟨*proof*⟩

4.2.6 Collinear points and h-lines

lemma *poincare-collinear3-poincare-line-general*:
assumes *poincare-collinear* $\{a,\ a1,\ a2\}$ $a1 \neq a2$ $a1 \neq$ *inversion a2*
shows $a \in$ *circline-set* (*poincare-line a1 a2*)
⟨*proof*⟩

lemma *poincare-line-poincare-collinear3-general*:
assumes $a \in$ *circline-set* (*poincare-line a1 a2*) $a1 \neq a2$
shows *poincare-collinear* $\{a,\ a1,\ a2\}$
⟨*proof*⟩

lemma *poincare-collinear3-poincare-lines-equal-general*:
assumes *poincare-collinear* $\{a,\ a1,\ a2\}$ $a \neq a1$ $a \neq a2$ $a \neq$ *inversion a1* $a \neq$ *inversion a2*
shows *poincare-line a a1* = *poincare-line a a2*
⟨*proof*⟩

4.2.7 Points collinear with 0_h

lemma *poincare-collinear-zero-iff*:
assumes *of-complex* $y' \in$ *unit-disc* **and** *of-complex* $z' \in$ *unit-disc* **and**
$y' \neq z'$ **and** $y' \neq 0$ **and** $z' \neq 0$
shows *poincare-collinear* $\{0_h,$ *of-complex* $y',$ *of-complex* $z'\}$ $\longleftrightarrow$
y'**cnj* z' = *cnj* y'*z' (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
⟨*proof*⟩

lemma *poincare-collinear-zero-polar-form*:
assumes *poincare-collinear* $\{0_h,$ *of-complex* $x,$ *of-complex* $y\}$ **and**
$x \neq 0$ **and** $y \neq 0$ **and** *of-complex* $x \in$ *unit-disc* **and** *of-complex* $y \in$ *unit-disc*
shows $\exists\ \varphi\ rx\ ry.\ x = cor\ rx * cis\ \varphi \wedge y = cor\ ry * cis\ \varphi \wedge rx \neq 0 \wedge ry \neq 0$
⟨*proof*⟩

end
theory *Poincare-Lines-Ideal-Points*
imports *Poincare-Lines*
begin

4.3 Ideal points of h-lines

Ideal points of an h-line are points where the h-line intersects the unit disc.

4.3.1 Calculation of ideal points

We decided to define ideal points constructively, i.e., we calculate the coordinates of ideal points for a given h-line explicitly. Namely, if the h-line is determined by A and B, the two intersection points are

$$\frac{B}{|B|^2}\left(-A \pm i \cdot \sqrt{|B|^2 - A^2}\right).$$

definition *calc-ideal-point1-cvec* :: *complex* ⇒ *complex* ⇒ *complex-vec* **where**
[*simp*]: *calc-ideal-point1-cvec A B* =
(*let discr* = *Re* ((*cmod B*)^2 − (*Re A*)^2) *in*
(*B*∗(−*A* − i∗*sqrt*(*discr*)), (*cmod B*)^2))

definition *calc-ideal-point2-cvec* :: *complex* ⇒ *complex* ⇒ *complex-vec* **where**
[*simp*]: *calc-ideal-point2-cvec A B* =
(*let discr* = *Re* ((*cmod B*)^2 − (*Re A*)^2) *in*
(*B*∗(−*A* + i∗*sqrt*(*discr*)), (*cmod B*)^2))

definition *calc-ideal-points-cmat-cvec* :: *complex-mat* ⇒ *complex-vec set* **where**
[*simp*]: *calc-ideal-points-cmat-cvec H* =
(*if is-poincare-line-cmat H then*
let (*A*, *B*, *C*, *D*) = *H*
in {*calc-ideal-point1-cvec A B*, *calc-ideal-point2-cvec A B*}
else
{(−*1*, *1*), (*1*, *1*)})

lift-definition *calc-ideal-points-clmat-hcoords* :: *circline-mat* ⇒ *complex-homo-coords set* **is** *calc-ideal-points-cmat-cvec*
⟨*proof*⟩

lift-definition *calc-ideal-points* :: *circline* ⇒ *complex-homo set* **is** *calc-ideal-points-clmat-hcoords*
⟨*proof*⟩

Correctness of the calculation

We show that for every h-line its two calculated ideal points are different and are on the intersection of that line and the unit circle.

Calculated ideal points are on the unit circle

lemma *calc-ideal-point-1-unit*:
assumes *is-real A* (*cmod B*)^2 > (*cmod A*)^2
assumes (*z1*, *z2*) = *calc-ideal-point1-cvec A B*
shows *z1* ∗ *cnj z1* = *z2* ∗ *cnj z2*
⟨*proof*⟩

lemma *calc-ideal-point-2-unit*:
assumes *is-real A* (*cmod B*)^2 > (*cmod A*)^2
assumes (*z1*, *z2*) = *calc-ideal-point2-cvec A B*
shows *z1* ∗ *cnj z1* = *z2* ∗ *cnj z2*
⟨*proof*⟩

lemma *calc-ideal-points-on-unit-circle*:
shows ∀ *z* ∈ *calc-ideal-points H*. *z* ∈ *circline-set unit-circle*
⟨*proof*⟩

Calculated ideal points are on the h-line

lemma *calc-ideal-point1-sq*:
assumes (*z1*, *z2*) = *calc-ideal-point1-cvec A B is-real A* (*cmod B*)^2 > (*cmod A*)^2
shows *z1* ∗ *cnj z1* + *z2* ∗ *cnj z2* = *2* ∗ (*B* ∗ *cnj B*)^2
⟨*proof*⟩

lemma *calc-ideal-point2-sq*:
assumes (*z1*, *z2*) = *calc-ideal-point2-cvec A B is-real A* (*cmod B*)^2 > (*cmod A*)^2
shows *z1* ∗ *cnj z1* + *z2* ∗ *cnj z2* = *2* ∗ (*B* ∗ *cnj B*)^2
⟨*proof*⟩

lemma *calc-ideal-point1-mix*:

assumes $(z1, z2) = calc\text{-}ideal\text{-}point1\text{-}cvec\ A\ B$ $is\text{-}real\ A$ $(cmod\ B)^2 > (cmod\ A)^2$
shows $B * cnj\ z1 * z2 + cnj\ B * z1 * cnj\ z2 = - 2 * A * (B * cnj\ B)^2$
⟨*proof*⟩

lemma *calc-ideal-point2-mix*:
assumes $(z1, z2) = calc\text{-}ideal\text{-}point2\text{-}cvec\ A\ B$ $is\text{-}real\ A$ $(cmod\ B)^2 > (cmod\ A)^2$
shows $B * cnj\ z1 * z2 + cnj\ B * z1 * cnj\ z2 = - 2 * A * (B * cnj\ B)^2$
⟨*proof*⟩

lemma *calc-ideal-point1-on-circline*:
assumes $(z1, z2) = calc\text{-}ideal\text{-}point1\text{-}cvec\ A\ B$ $is\text{-}real\ A$ $(cmod\ B)^2 > (cmod\ A)^2$
shows $A*z1*cnj\ z1 + B*cnj\ z1*z2 + cnj\ B*z1*cnj\ z2 + A*z2*cnj\ z2 = 0$ (**is** *?lhs = 0*)
⟨*proof*⟩

lemma *calc-ideal-point2-on-circline*:
assumes $(z1, z2) = calc\text{-}ideal\text{-}point2\text{-}cvec\ A\ B$ $is\text{-}real\ A$ $(cmod\ B)^2 > (cmod\ A)^2$
shows $A*z1*cnj\ z1 + B*cnj\ z1*z2 + cnj\ B*z1*cnj\ z2 + A*z2*cnj\ z2 = 0$ (**is** *?lhs = 0*)
⟨*proof*⟩

lemma *calc-ideal-points-on-circline*:
assumes *is-poincare-line H*
shows ∀ *z* ∈ *calc-ideal-points H*. *z* ∈ *circline-set H*
⟨*proof*⟩

Calculated ideal points of an h-line are different

lemma *calc-ideal-points-cvec-different* [*simp*]:
assumes $(cmod\ B)^2 > (cmod\ A)^2$ *is-real A*
shows ¬ (*calc-ideal-point1-cvec A B* $\approx_v$ *calc-ideal-point2-cvec A B*)
⟨*proof*⟩

lemma *calc-ideal-points-different*:
assumes *is-poincare-line H*
shows ∃ *i1* ∈ (*calc-ideal-points H*). ∃ *i2* ∈ (*calc-ideal-points H*). *i1* ≠ *i2*
⟨*proof*⟩

lemma *two-calc-ideal-points* [*simp*]:
assumes *is-poincare-line H*
shows *card* (*calc-ideal-points H*) = *2*
⟨*proof*⟩

4.3.2 Ideal points

Next we give a genuine definition of ideal points – these are the intersections of the h-line with the unit circle

definition *ideal-points* :: *circline* ⇒ *complex-homo set* **where**
ideal-points H = *circline-intersection H unit-circle*

Ideal points are on the unit circle and on the h-line

lemma *ideal-points-on-unit-circle*:
shows ∀ *z* ∈ *ideal-points H*. *z* ∈ *circline-set unit-circle*
⟨*proof*⟩

lemma *ideal-points-on-circline*:
shows ∀ *z* ∈ *ideal-points H*. *z* ∈ *circline-set H*
⟨*proof*⟩

For each h-line there are exactly two ideal points

lemma *two-ideal-points*:
assumes *is-poincare-line H*
shows *card* (*ideal-points H*) = *2*
⟨*proof*⟩

They are exactly the two points that our calculation finds

lemma *ideal-points-unique*:
assumes *is-poincare-line H*

shows *ideal-points H* = *calc-ideal-points H*
⟨*proof*⟩

For each h-line we can obtain two different ideal points

lemma *obtain-ideal-points*:
assumes *is-poincare-line H*
obtains *i1 i2* **where** *i1* ≠ *i2 ideal-points H* = {*i1*, *i2*}
⟨*proof*⟩

Ideal points of each h-line constructed from two points in the disc are different than those two points

lemma *ideal-points-different*:
assumes *u* ∈ *unit-disc v* ∈ *unit-disc u* ≠ *v*
assumes *ideal-points* (*poincare-line u v*) = {*i1*, *i2*}
shows *i1* ≠ *i2 u* ≠ *i1 u* ≠ *i2 v* ≠ *i1 v* ≠ *i2*
⟨*proof*⟩

H-line is uniquely determined by its ideal points

lemma *ideal-points-line-unique*:
assumes *is-poincare-line H ideal-points H* = {*i1*, *i2*}
shows *H* = *poincare-line i1 i2*
⟨*proof*⟩

Ideal points of some special h-lines

Ideal points of *x-axis*

lemma *ideal-points-x-axis*
[*simp*]: *ideal-points x-axis* = {*of-complex* (−*1*), *of-complex 1*}
⟨*proof*⟩

Ideal points are proportional vectors only if h-line is a line segment passing trough zero

lemma *ideal-points-proportional*:
assumes *is-poincare-line H ideal-points H* = {*i1*, *i2*} *to-complex i1* = *cor k* ∗ *to-complex i2*
shows 0_h ∈ *circline-set H*
⟨*proof*⟩

Transformations of ideal points

Möbius transformations that fix the unit disc when acting on h-lines map their ideal points to ideal points.

lemma *ideal-points-moebius-circline* [*simp*]:
assumes *unit-circle-fix M is-poincare-line H*
shows *ideal-points* (*moebius-circline M H*) = (*moebius-pt M*) ‘ (*ideal-points H*) (**is** *?I′* = *?M* ‘ *?I*)
⟨*proof*⟩

lemma *ideal-points-poincare-line-moebius* [*simp*]:
assumes *unit-disc-fix M u* ∈ *unit-disc v* ∈ *unit-disc u* ≠ *v*
assumes *ideal-points* (*poincare-line u v*) = {*i1*, *i2*}
shows *ideal-points* (*poincare-line* (*moebius-pt M u*) (*moebius-pt M v*)) = {*moebius-pt M i1*, *moebius-pt M i2*}
⟨*proof*⟩

Conjugation also maps ideal points to ideal points

lemma *ideal-points-conjugate* [*simp*]:
assumes *is-poincare-line H*
shows *ideal-points* (*conjugate-circline H*) = *conjugate* ‘ (*ideal-points H*) (**is** *?I′* = *?M* ‘ *?I*)
⟨*proof*⟩

lemma *ideal-points-poincare-line-conjugate* [*simp*]:
assumes*u* ∈ *unit-disc v* ∈ *unit-disc u* ≠ *v*
assumes *ideal-points* (*poincare-line u v*) = {*i1*, *i2*}
shows *ideal-points* (*poincare-line* (*conjugate u*) (*conjugate v*)) = {*conjugate i1*, *conjugate i2*}
⟨*proof*⟩

end
theory *Poincare-Distance*
imports *Poincare-Lines-Ideal-Points Hyperbolic-Functions*
begin

5 H-distance in the Poincaré model

Informally, the *h-distance* between the two h-points is defined as the absolute value of the logarithm of the cross ratio between those two points and the two ideal points.

abbreviation *Re-cross-ratio* **where** *Re-cross-ratio z u v w* ≡ *Re* (*to-complex* (*cross-ratio z u v w*))

definition *calc-poincare-distance* :: *complex-homo* ⇒ *complex-homo* ⇒ *complex-homo* ⇒ *complex-homo* ⇒ *real* **where**
[*simp*]: *calc-poincare-distance u i1 v i2* = *abs* (*ln* (*Re-cross-ratio u i1 v i2*))

definition *poincare-distance-pred* :: *complex-homo* ⇒ *complex-homo* ⇒ *real* ⇒ *bool* **where**
[*simp*]: *poincare-distance-pred u v d* ⟷
(*u* = *v* ∧ *d* = *0*) ∨ (*u* ≠ *v* ∧ (∀ *i1 i2*. *ideal-points* (*poincare-line u v*) = {*i1*, *i2*} ⟶ *d* = *calc-poincare-distance u i1 v i2*))

definition *poincare-distance* :: *complex-homo* ⇒ *complex-homo* ⇒ *real* **where**
poincare-distance u v = (*THE d*. *poincare-distance-pred u v d*)

We shown that the described cross-ratio is always finite, positive real number.

lemma *distance-cross-ratio-real-positive*:
assumes *u* ∈ *unit-disc* **and** *v* ∈ *unit-disc* **and** *u* ≠ *v*
shows ∀ *i1 i2*. *ideal-points* (*poincare-line u v*) = {*i1*, *i2*} ⟶
cross-ratio u i1 v i2 ≠ ∞_h ∧ *is-real* (*to-complex* (*cross-ratio u i1 v i2*)) ∧ *Re-cross-ratio u i1 v i2* > *0*
(**is** *?P u v*)
⟨*proof*⟩

Next we can show that for every different points from the unit disc there is exactly one number that satisfies the h-distance predicate.

lemma *distance-unique*:
assumes *u* ∈ *unit-disc* **and** *v* ∈ *unit-disc*
shows ∃! *d*. *poincare-distance-pred u v d*
⟨*proof*⟩

lemma *poincare-distance-satisfies-pred* [*simp*]:
assumes *u* ∈ *unit-disc* **and** *v* ∈ *unit-disc*
shows *poincare-distance-pred u v* (*poincare-distance u v*)
⟨*proof*⟩

lemma *poincare-distance-I*:
assumes *u* ∈ *unit-disc* **and** *v* ∈ *unit-disc* **and** *u* ≠ *v* **and** *ideal-points* (*poincare-line u v*) = {*i1*, *i2*}
shows *poincare-distance u v* = *calc-poincare-distance u i1 v i2*
⟨*proof*⟩

lemma *poincare-distance-refl* [*simp*]:
assumes *u* ∈ *unit-disc*
shows *poincare-distance u u* = *0*
⟨*proof*⟩

Unit disc preserving Möbius transformations preserve h-distance.

lemma *unit-disc-fix-preserve-poincare-distance* [*simp*]:
assumes *unit-disc-fix M* **and** *u* ∈ *unit-disc* **and** *v* ∈ *unit-disc*
shows *poincare-distance* (*moebius-pt M u*) (*moebius-pt M v*) = *poincare-distance u v*
⟨*proof*⟩

Knowing ideal points for x-axis, we can easily explicitly calculate distances.

lemma *poincare-distance-x-axis-x-axis*:
assumes *x* ∈ *unit-disc* **and** *y* ∈ *unit-disc* **and** *x* ∈ *circline-set x-axis* **and** *y* ∈ *circline-set x-axis*
shows *poincare-distance x y* =
(*let* x' = *to-complex x*; y' = *to-complex y*
in abs (*ln* (*Re* ((($1 + x'$) ∗ ($1 - y'$)) / (($1 - x'$) ∗ ($1 + y'$))))))
⟨*proof*⟩

lemma *poincare-distance-zero-x-axis*:
assumes *x* ∈ *unit-disc* **and** *x* ∈ *circline-set x-axis*
shows *poincare-distance* 0_h *x* = (*let* x' = *to-complex x in abs* (*ln* (*Re* (($1 - x'$) / ($1 + x'$)))))

⟨proof⟩

lemma *poincare-distance-zero*:
assumes $x \in$ *unit-disc*
shows *poincare-distance* 0_h x = (*let* x' = *to-complex* x *in abs* (*ln* (*Re* ((*1* − *cmod* x') / (*1* + *cmod* x'))))) (**is** *?P* x)
⟨proof⟩

lemma *poincare-distance-zero-opposite* [*simp*]:
assumes *of-complex* $z \in$ *unit-disc*
shows *poincare-distance* 0_h (*of-complex* (− z)) = *poincare-distance* 0_h (*of-complex* z)
⟨proof⟩

5.1 Distance explicit formula

Instead of the h-distance itself, very frequently its hyperbolic cosine is analyzed.

abbreviation *cosh-dist u v* ≡ *cosh* (*poincare-distance u v*)

lemma *cosh-poincare-distance-cross-ratio-average*:
assumes $u \in$ *unit-disc* $v \in$ *unit-disc* $u \neq v$ *ideal-points* (*poincare-line u v*) = {*i1*, *i2*}
shows *cosh-dist u v* =
((*Re-cross-ratio u i1 v i2*) + (*Re-cross-ratio v i1 u i2*)) / *2*
⟨proof⟩

definition *poincare-distance-formula'* :: *complex* ⇒ *complex* ⇒ *real* **where**
[*simp*]: *poincare-distance-formula' u v* = *1* + *2* ∗ ((*cmod* (*u* − *v*))^2 / ((*1* − (*cmod u*)^2) ∗ (*1* − (*cmod v*)^2)))

Next we show that the following formula expresses h-distance between any two h-points (note that the ideal points do not figure anymore).

definition *poincare-distance-formula* :: *complex* ⇒ *complex* ⇒ *real* **where**
[*simp*]: *poincare-distance-formula u v* = *arcosh* (*poincare-distance-formula' u v*)

lemma *blaschke-preserve-distance-formula* [*simp*]:
assumes *of-complex* $k \in$ *unit-disc* $u \in$ *unit-disc* $v \in$ *unit-disc*
shows *poincare-distance-formula* (*to-complex* (*moebius-pt* (*blaschke k*) *u*)) (*to-complex* (*moebius-pt* (*blaschke k*) *v*)) =
poincare-distance-formula (*to-complex u*) (*to-complex v*)
⟨proof⟩

To prove the equivalence between the h-distance definition and the distance formula, we shall employ the without loss of generality principle. Therefore, we must show that the distance formula is preserved by h-isometries.

Rotation preserve *poincare-distance-formula*.

lemma *rotation-preserve-distance-formula* [*simp*]:
assumes $u \in$ *unit-disc* $v \in$ *unit-disc*
shows *poincare-distance-formula* (*to-complex* (*moebius-pt* (*moebius-rotation* φ) *u*)) (*to-complex* (*moebius-pt* (*moebius-rotation* φ) *v*)) =
poincare-distance-formula (*to-complex u*) (*to-complex v*)
⟨proof⟩

Unit disc fixing Möbius preserve *poincare-distance-formula*.

lemma *unit-disc-fix-preserve-distance-formula* [*simp*]:
assumes *unit-disc-fix M* $u \in$ *unit-disc* $v \in$ *unit-disc*
shows *poincare-distance-formula* (*to-complex* (*moebius-pt M u*)) (*to-complex* (*moebius-pt M v*)) =
poincare-distance-formula (*to-complex u*) (*to-complex v*) (**is** *?P' u v M*)
⟨proof⟩

The equivalence between the two h-distance representations.

lemma *poincare-distance-formula*:
assumes $u \in$ *unit-disc* **and** $v \in$ *unit-disc*
shows *poincare-distance u v* = *poincare-distance-formula* (*to-complex u*) (*to-complex v*) (**is** *?P u v*)
⟨proof⟩

Some additional properties proved easily using the distance formula.

poincare-distance is symmetric.

lemma *poincare-distance-sym*:
assumes $u \in$ *unit-disc* **and** $v \in$ *unit-disc*
shows *poincare-distance u v* = *poincare-distance v u*
⟨*proof*⟩

lemma *poincare-distance-formula′-ge-1*:
assumes $u \in$ *unit-disc* **and** $v \in$ *unit-disc*
shows $1 \leq$ *poincare-distance-formula′* (*to-complex u*) (*to-complex v*)
⟨*proof*⟩

poincare-distance is non-negative.

lemma *poincare-distance-ge0*:
assumes $u \in$ *unit-disc* **and** $v \in$ *unit-disc*
shows *poincare-distance u v* ≥ 0
⟨*proof*⟩

lemma *cosh-dist*:
assumes $u \in$ *unit-disc* **and** $v \in$ *unit-disc*
shows *cosh-dist u v* = *poincare-distance-formula′* (*to-complex u*) (*to-complex v*)
⟨*proof*⟩

poincare-distance is zero only if the two points are equal.

lemma *poincare-distance-eq-0-iff*:
assumes $u \in$ *unit-disc* **and** $v \in$ *unit-disc*
shows *poincare-distance u v* = $0 \longleftrightarrow u = v$
⟨*proof*⟩

Conjugate preserve *poincare-distance-formula*.

lemma *conjugate-preserve-poincare-distance* [*simp*]:
assumes $u \in$ *unit-disc* **and** $v \in$ *unit-disc*
shows *poincare-distance* (*conjugate u*) (*conjugate v*) = *poincare-distance u v*
⟨*proof*⟩

5.2 Existence and uniqueness of points with a given distance

lemma *ex-x-axis-poincare-distance-negative′*:
fixes *d* :: *real*
assumes $d \geq 0$
shows *let z* = $(1 - exp\ d) / (1 + exp\ d)$
in is-real z ∧ *Re z* ≤ 0 ∧ *Re z* > -1 ∧
of-complex z ∈ *unit-disc* ∧ *of-complex z* ∈ *circline-set x-axis* ∧
poincare-distance 0_h (*of-complex z*) = *d*
⟨*proof*⟩

lemma *ex-x-axis-poincare-distance-negative*:
assumes $d \geq 0$
shows ∃ *z*. *is-real z* ∧ *Re z* ≤ 0 ∧ *Re z* > -1 ∧
of-complex z ∈ *unit-disc* ∧ *of-complex z* ∈ *circline-set x-axis* ∧
poincare-distance 0_h (*of-complex z*) = *d* (**is** ∃ *z*. *?P z*)
⟨*proof*⟩

For each real number *d* there is exactly one point on the positive x-axis such that h-distance between 0 and that point is *d*.

lemma *unique-x-axis-poincare-distance-negative*:
assumes $d \geq 0$
shows ∃! *z*. *is-real z* ∧ *Re z* ≤ 0 ∧ *Re z* > -1 ∧
poincare-distance 0_h (*of-complex z*) = *d* (**is** ∃! *z*. *?P z*)
⟨*proof*⟩

lemma *ex-x-axis-poincare-distance-positive*:
assumes $d \geq 0$
shows ∃ *z*. *is-real z* ∧ *Re z* ≥ 0 ∧ *Re z* < 1 ∧
of-complex z ∈ *unit-disc* ∧ *of-complex z* ∈ *circline-set x-axis* ∧
poincare-distance 0_h (*of-complex z*) = *d* (**is** ∃ *z*. *is-real z* ∧ *Re z* ≥ 0 ∧ *Re z* < 1 ∧ *?P z*)
⟨*proof*⟩

lemma *unique-x-axis-poincare-distance-positive*:
assumes $d \geq 0$
shows $\exists!\ z.\ is\text{-}real\ z \wedge Re\ z \geq 0 \wedge Re\ z < 1 \wedge$
$poincare\text{-}distance\ 0_h\ (of\text{-}complex\ z) = d$ (**is** $\exists!\ z.\ is\text{-}real\ z \wedge Re\ z \geq 0 \wedge Re\ z < 1 \wedge\ ?P\ z$)
⟨*proof*⟩

Equal distance implies that segments are isometric - this means that congruence could be defined either by two segments having the same distance or by requiring existence of an isometry that maps one segment to the other.

lemma *poincare-distance-eq-ex-moebius*:
assumes *in-disc*: $u \in unit\text{-}disc$ **and** $v \in unit\text{-}disc$ **and** $u' \in unit\text{-}disc$ **and** $v' \in unit\text{-}disc$
assumes $poincare\text{-}distance\ u\ v = poincare\text{-}distance\ u'\ v'$
shows $\exists\ M.\ unit\text{-}disc\text{-}fix\ M \wedge moebius\text{-}pt\ M\ u = u' \wedge moebius\text{-}pt\ M\ v = v'$ (**is** $?P'\ u\ v\ u'\ v'$)
⟨*proof*⟩

lemma *unique-midpoint-x-axis*:
assumes *x*: $is\text{-}real\ x\ {-1} < Re\ x\ Re\ x < 1$ **and**
y: $is\text{-}real\ y\ {-1} < Re\ y\ Re\ y < 1$ **and**
$x \neq y$
shows $\exists!\ z.\ {-1} < Re\ z \wedge Re\ z < 1 \wedge is\text{-}real\ z \wedge poincare\text{-}distance\ (of\text{-}complex\ z)\ (of\text{-}complex\ x) = poincare\text{-}distance\ (of\text{-}complex\ z)\ (of\text{-}complex\ y)$ (**is** $\exists!\ z.\ ?R\ z\ (of\text{-}complex\ x)\ (of\text{-}complex\ y)$)
⟨*proof*⟩

5.3 Triangle inequality

lemma *poincare-distance-formula-zero-sum*:
assumes $u \in unit\text{-}disc$ **and** $v \in unit\text{-}disc$
shows $poincare\text{-}distance\ u\ 0_h + poincare\text{-}distance\ 0_h\ v =$
$(let\ u' = cmod\ (to\text{-}complex\ u);\ v' = cmod\ (to\text{-}complex\ v)$
$in\ arcosh\ (((1 + u'^2) * (1 + v'^2) + 4 * u' * v') / ((1 - u'^2) * (1 - v'^2))))$
⟨*proof*⟩

lemma *poincare-distance-triangle-inequality*:
assumes $u \in unit\text{-}disc$ **and** $v \in unit\text{-}disc$ **and** $w \in unit\text{-}disc$
shows $poincare\text{-}distance\ u\ v + poincare\text{-}distance\ v\ w \geq poincare\text{-}distance\ u\ w$ (**is** $?P'\ u\ v\ w$)
⟨*proof*⟩

end
theory *Poincare-Circles*
imports *Poincare-Distance*
begin

6 H-circles in the Poincaré model

Circles consist of points that are at the same distance from the center.

definition *poincare-circle* :: *complex-homo* ⇒ *real* ⇒ *complex-homo set* **where**
$poincare\text{-}circle\ z\ r = \{z'.\ z' \in unit\text{-}disc \wedge poincare\text{-}distance\ z\ z' = r\}$

Each h-circle in the Poincaré model is represented by an Euclidean circle in the model — the center and radius of that euclidean circle are determined by the following formulas.

definition *poincare-circle-euclidean* :: *complex-homo* ⇒ *real* ⇒ *euclidean-circle* **where**
$poincare\text{-}circle\text{-}euclidean\ z\ r =$
$(let\ R = (cosh\ r - 1) / 2;$
$z' = to\text{-}complex\ z;$
$cz = 1 - (cmod\ z')^2;$
$k = cz * R + 1$
$in\ (z' / k,\ cz * sqrt(R * (R + 1)) / k))$

That Euclidean circle has a positive radius and is always fully within the disc.

lemma *poincare-circle-in-disc*:
assumes $r > 0$ **and** $z \in unit\text{-}disc$ **and** $(ze, re) = poincare\text{-}circle\text{-}euclidean\ z\ r$
shows $cmod\ ze < 1\ re > 0\ \forall\ x \in circle\ ze\ re.\ cmod\ x < 1$
⟨*proof*⟩

The connection between the points on the h-circle and its corresponding Euclidean circle.

lemma *poincare-circle-is-euclidean-circle*:
  assumes $z \in$ *unit-disc* **and** $r > 0$
  shows *let* (*Ze*, *Re*) = *poincare-circle-euclidean z r*
    in of-complex ' (*circle Ze Re*) = *poincare-circle z r*
⟨*proof*⟩

6.1 Intersection of circles in special positions

Two h-circles centered at the x-axis intersect at mutually conjugate points

lemma *intersect-poincare-circles-x-axis*:
  assumes *z*: *is-real z1* **and** *is-real z2* **and** $r1 > 0$ **and** $r2 > 0$ **and**
    $-1 <$ *Re z1* **and** *Re z1* < 1 **and** $-1 <$ *Re z2* **and** *Re z2* < 1 **and**
    $z1 \neq z2$
  assumes *x1*: $x1 \in$ *poincare-circle* (*of-complex z1*) *r1* ∩ *poincare-circle* (*of-complex z2*) *r2* **and**
    x2: $x2 \in$ *poincare-circle* (*of-complex z1*) *r1* ∩ *poincare-circle* (*of-complex z2*) *r2* **and**
    $x1 \neq x2$
  shows *x1* = *conjugate x2*
⟨*proof*⟩

Two h-circles of the same radius centered at mutually conjugate points intersect at the x-axis

lemma *intersect-poincare-circles-conjugate-centers*:
  assumes *in-disc*: $z1 \in$ *unit-disc* $z2 \in$ *unit-disc* **and**
    $z1 \neq z2$ **and** *z1* = *conjugate z2* **and** $r > 0$ **and**
    u: $u \in$ *poincare-circle z1 r* ∩ *poincare-circle z2 r*
  shows *is-real* (*to-complex u*)
⟨*proof*⟩

6.2 Congruent triangles

For every pair of triangles such that its three pairs of sides are pairwise equal there is an h-isometry (a unit disc preserving Möbius transform, eventually composed with a conjugation) that maps one triangle onto the other.

lemma *unit-disc-fix-f-congruent-triangles*:
  assumes
    in-disc: $u \in$ *unit-disc* $v \in$ *unit-disc* $w \in$ *unit-disc* **and**
    in-disc': $u' \in$ *unit-disc* $v' \in$ *unit-disc* $w' \in$ *unit-disc* **and**
    d: *poincare-distance u v* = *poincare-distance u' v'*
      poincare-distance v w = *poincare-distance v' w'*
      poincare-distance u w = *poincare-distance u' w'*
  shows
    ∃ *M*. *unit-disc-fix-f M* ∧ *M u* = *u'* ∧ *M v* = *v'* ∧ *M w* = *w'*
⟨*proof*⟩

end
theory *Poincare-Between*
  imports *Poincare-Distance*
begin

7 H-betweenness in the Poincaré model

7.1 H-betwenness expressed by a cross-ratio

The point v is h-between u and w if the cross-ratio between the pairs u and w and v and inverse of v is real and negative.

definition *poincare-between* :: *complex-homo* ⇒ *complex-homo* ⇒ *complex-homo* ⇒ *bool* **where**
  poincare-between u v w ⟷
    u = *v* ∨ *v* = *w* ∨
    (*let cr* = *cross-ratio u v w* (*inversion v*)
      in is-real (*to-complex cr*) ∧ *Re* (*to-complex cr*) < 0)

7.1.1 H-betwenness is preserved by h-isometries

Since they preserve cross-ratio and inversion, h-isometries (unit disc preserving Möbius transformations and conjugation) preserve h-betweeness.

lemma *unit-disc-fix-moebius-preserve-poincare-between* [*simp*]:
  assumes *unit-disc-fix M* **and** $u \in$ *unit-disc* **and** $v \in$ *unit-disc* **and** $w \in$ *unit-disc*
  shows *poincare-between* (*moebius-pt M u*) (*moebius-pt M v*) (*moebius-pt M w*) $\longleftrightarrow$
    poincare-between u v w
⟨*proof*⟩

lemma *conjugate-preserve-poincare-between* [*simp*]:
  assumes $u \in$ *unit-disc* **and** $v \in$ *unit-disc* **and** $w \in$ *unit-disc*
  shows *poincare-between* (*conjugate u*) (*conjugate v*) (*conjugate w*) $\longleftrightarrow$
    poincare-between u v w
⟨*proof*⟩

7.1.2 Some elementary properties of h-betwenness

lemma *poincare-between-nonstrict* [*simp*]:
  shows *poincare-between u u v* **and** *poincare-between u v v*
  ⟨*proof*⟩

lemma *poincare-between-sandwich*:
  assumes $u \in$ *unit-disc* **and** $v \in$ *unit-disc*
  assumes *poincare-between u v u*
  shows $u = v$
⟨*proof*⟩

lemma *poincare-between-rev*:
  assumes $u \in$ *unit-disc* **and** $v \in$ *unit-disc* **and** $w \in$ *unit-disc*
  shows *poincare-between u v w* $\longleftrightarrow$ *poincare-between w v u*
  ⟨*proof*⟩

7.1.3 H-betwenness and h-collinearity

Three points can be in an h-between relation only when they are h-collinear.

lemma *poincare-between-poincare-collinear* [*simp*]:
  assumes *in-disc*: $u \in$ *unit-disc* $v \in$ *unit-disc* $w \in$ *unit-disc*
  assumes *betw*: *poincare-between u v w*
  shows *poincare-collinear* $\{u, v, w\}$
⟨*proof*⟩

lemma *poincare-between-poincare-line-uvz*:
  assumes $u \neq v$ **and** $u \in$ *unit-disc* **and** $v \in$ *unit-disc* **and**
    $z \in$ *unit-disc* **and** *poincare-between u v z*
  shows $z \in$ *circline-set* (*poincare-line u v*)
  ⟨*proof*⟩

lemma *poincare-between-poincare-line-uzv*:
  assumes $u \neq v$ **and** $u \in$ *unit-disc* **and** $v \in$ *unit-disc* **and**
    $z \in$ *unit-disc poincare-between u z v*
  shows $z \in$ *circline-set* (*poincare-line u v*)
  ⟨*proof*⟩

7.1.4 H-betweeness on Euclidean segments

If the three points lie on an h-line that is a Euclidean line (e.g., if it contains zero), h-betweenness can be characterized much simpler than in the definition.

lemma *poincare-between-x-axis-u0v*:
  assumes *is-real* u' **and** $u' \neq 0$ **and** $v' \neq 0$
  shows *poincare-between* (*of-complex* u') 0_h (*of-complex* v') $\longleftrightarrow$ *is-real* $v' \wedge$ *Re* $u' *$ *Re* $v' < 0$
⟨*proof*⟩

lemma *poincare-between-u0v*:
  assumes $u \in$ *unit-disc* **and** $v \in$ *unit-disc* **and** $u \neq 0_h$ **and** $v \neq 0_h$

shows *poincare-between u* 0_h *v* $\longleftrightarrow$ ($\exists$ *k* $<$ *0*. *to-complex u* $=$ *cor k* $*$ *to-complex v*) (**is** *?P u v*)
⟨*proof*⟩

lemma *poincare-between-u0v-polar-form*:
assumes $x \in$ *unit-disc* **and** $y \in$ *unit-disc* **and** $x \neq 0_h$ **and** $y \neq 0_h$ **and**
to-complex x $=$ *cor rx* $*$ *cis* φ *to-complex y* $=$ *cor ry* $*$ *cis* φ
shows *poincare-between x* 0_h *y* $\longleftrightarrow$ *rx* $*$ *ry* $<$ *0* (**is** *?P x y rx ry*)
⟨*proof*⟩

lemma *poincare-between-x-axis-0uv*:
fixes *x y* :: *real*
assumes $-1 < x$ **and** $x < 1$ **and** $x \neq 0$
assumes $-1 < y$ **and** $y < 1$ **and** $y \neq 0$
shows *poincare-between* 0_h (*of-complex x*) (*of-complex y*) $\longleftrightarrow$
($x < 0 \wedge y < 0 \wedge y \leq x$) $\vee$ ($x > 0 \wedge y > 0 \wedge x \leq y$) (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
⟨*proof*⟩

lemma *poincare-between-0uv*:
assumes $u \in$ *unit-disc* **and** $v \in$ *unit-disc* **and** $u \neq 0_h$ **and** $v \neq 0_h$
shows *poincare-between* 0_h *u v* $\longleftrightarrow$
(*let u'* $=$ *to-complex u*; *v'* $=$ *to-complex v in Arg u'* $=$ *Arg v'* $\wedge$ *cmod u'* $\leq$ *cmod v'*) (**is** *?P u v*)
⟨*proof*⟩

lemma *poincare-between-y-axis-0uv*:
fixes *x y* :: *real*
assumes $-1 < x$ **and** $x < 1$ **and** $x \neq 0$
assumes $-1 < y$ **and** $y < 1$ **and** $y \neq 0$
shows *poincare-between* 0_h (*of-complex* (i $*$ *x*)) (*of-complex* (i $*$ *y*)) $\longleftrightarrow$
($x < 0 \wedge y < 0 \wedge y \leq x$) $\vee$ ($x > 0 \wedge y > 0 \wedge x \leq y$) (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
⟨*proof*⟩

lemma *poincare-between-x-axis-uvw*:
fixes *x y z* :: *real*
assumes $-1 < x$ **and** $x < 1$
assumes $-1 < y$ **and** $y < 1$ **and** $y \neq x$
assumes $-1 < z$ **and** $z < 1$ **and** $z \neq x$
shows *poincare-between* (*of-complex x*) (*of-complex y*) (*of-complex z*) $\longleftrightarrow$
($y < x \wedge z < x \wedge z \leq y$) $\vee$ ($y > x \wedge z > x \wedge y \leq z$) (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
⟨*proof*⟩

7.1.5 H-betweenness and h-collinearity

For three h-collinear points at least one of the three possible h-betweeness relations must hold.

lemma *poincare-collinear3-between*:
assumes $u \in$ *unit-disc* **and** $v \in$ *unit-disc* **and** $w \in$ *unit-disc*
assumes *poincare-collinear* {*u*, *v*, *w*}
shows *poincare-between u v w* $\vee$ *poincare-between u w v* $\vee$ *poincare-between v u w* (**is** *?P' u v w*)
⟨*proof*⟩

lemma *poincare-collinear3-iff*:
assumes $u \in$ *unit-disc* $v \in$ *unit-disc* $w \in$ *unit-disc*
shows *poincare-collinear* {*u*, *v*, *w*} $\longleftrightarrow$ *poincare-between u v w* $\vee$ *poincare-between v u w* $\vee$ *poincare-between v w u*
⟨*proof*⟩

7.2 Some properties of betweenness

lemma *poincare-between-transitivity*:
assumes $a \in$ *unit-disc* **and** $x \in$ *unit-disc* **and** $b \in$ *unit-disc* **and** $y \in$ *unit-disc* **and**
poincare-between a x b **and** *poincare-between a b y*
shows *poincare-between x b y*
⟨*proof*⟩

7.3 Poincare between - sum distances

Another possible definition of the h-betweenness relation is given in terms of h-distances between pairs of points. We prove it as a characterization equivalent to our cross-ratio based definition.

lemma *poincare-between-sum-distances-x-axis-u0v*:
  assumes *of-complex* $u' \in$ *unit-disc of-complex* $v' \in$ *unit-disc*
  assumes *is-real* u' $u' \neq 0$ $v' \neq 0$
  shows *poincare-distance* (*of-complex* u') 0_h + *poincare-distance* 0_h (*of-complex* v') = *poincare-distance* (*of-complex* u') (*of-complex* v') $\longleftrightarrow$
    is-real $v' \wedge$ *Re* $u' *$ *Re* $v' < 0$ (**is** *?P* u' v' $\longleftrightarrow$ *?Q* u' v')
⟨*proof*⟩

Different proof of the previous theorem relying on the cross-ratio definition, and not the distance formula. We suppose that this could be also used to prove the triangle inequality.

lemma *poincare-between-sum-distances-x-axis-u0v-different-proof*:
  assumes *of-complex* $u' \in$ *unit-disc of-complex* $v' \in$ *unit-disc*
  assumes *is-real* u' $u' \neq 0$ $v' \neq 0$ *is-real* v'
  shows *poincare-distance* (*of-complex* u') 0_h + *poincare-distance* 0_h (*of-complex* v') = *poincare-distance* (*of-complex* u') (*of-complex* v') $\longleftrightarrow$
    Re $u' *$ *Re* $v' < 0$ (**is** *?P* u' v' $\longleftrightarrow$ *?Q* u' v')
⟨*proof*⟩

lemma *poincare-between-sum-distances*:
  assumes $u \in$ *unit-disc* **and** $v \in$ *unit-disc* **and** $w \in$ *unit-disc*
  shows *poincare-between* u v w $\longleftrightarrow$
    poincare-distance u v + *poincare-distance* v w = *poincare-distance* u w (**is** *?P'* u v w)
⟨*proof*⟩

7.4 Some more properties of h-betweenness.

Some lemmas proved earlier are proved almost directly using the sum of distances characterization.

lemma *unit-disc-fix-moebius-preserve-poincare-between'*:
  assumes *unit-disc-fix* M **and** $u \in$ *unit-disc* **and** $v \in$ *unit-disc* **and** $w \in$ *unit-disc*
  shows *poincare-between* (*moebius-pt* M u) (*moebius-pt* M v) (*moebius-pt* M w) $\longleftrightarrow$
    poincare-between u v w
  ⟨*proof*⟩

lemma *conjugate-preserve-poincare-between'*:
  assumes $u \in$ *unit-disc* $v \in$ *unit-disc* $w \in$ *unit-disc*
  shows *poincare-between* (*conjugate* u) (*conjugate* v) (*conjugate* w) $\longleftrightarrow$ *poincare-between* u v w
  ⟨*proof*⟩

There is a unique point on a ray on the given distance from the given starting point

lemma *unique-poincare-distance-on-ray*:
  assumes $d \geq 0$ $u \neq v$ $u \in$ *unit-disc* $v \in$ *unit-disc*
  assumes $y \in$ *unit-disc poincare-distance* u y = d *poincare-between* u v y
  assumes $z \in$ *unit-disc poincare-distance* u z = d *poincare-between* u v z
  shows $y = z$
⟨*proof*⟩

end
theory *Poincare-Lines-Axis-Intersections*
  imports *Poincare-Between*
begin

8 Intersection of h-lines with the x-axis in the Poincaré model

8.1 Betweeness of x-axis intersection

The intersection point of the h-line determined by points u and v and the x-axis is between u and v, then u and v are in the opposite half-planes (one must be in the upper, and the other one in the lower half-plane).

lemma *poincare-between-x-axis-intersection*:
  assumes $u \in$ *unit-disc* **and** $v \in$ *unit-disc* **and** $z \in$ *unit-disc* **and** $u \neq v$
  assumes $u \notin$ *circline-set x-axis* **and** $v \notin$ *circline-set x-axis*
  assumes $z \in$ *circline-set* (*poincare-line* u v) $\cap$ *circline-set x-axis*
  shows *poincare-between* u z v $\longleftrightarrow$ *Arg* (*to-complex* u) $*$ *Arg* (*to-complex* v) < 0
⟨*proof*⟩

8.2 Check if an h-line intersects the x-axis

lemma *x-axis-intersection-equation*:
assumes
$H = mk\text{-}circline\ A\ B\ C\ D$ **and**
$(A,\ B,\ C,\ D) \in hermitean\text{-}nonzero$
shows $of\text{-}complex\ z \in circline\text{-}set\ x\text{-}axis \cap circline\text{-}set\ H \longleftrightarrow$
$A{*}z^2 + 2{*}Re\ B{*}z + D = 0 \wedge is\text{-}real\ z$ (**is** *?lhs* ⟷ *?rhs*)
⟨*proof*⟩

Check if an h-line intersects x-axis within the unit disc - this could be generalized to checking if an arbitrary circline intersects the x-axis, but we do not need that.

definition *intersects-x-axis-cmat* :: *complex-mat* ⇒ *bool* **where**
[*simp*]: $intersects\text{-}x\text{-}axis\text{-}cmat\ H = (let\ (A,\ B,\ C,\ D) = H\ in\ A = 0 \vee (Re\ B)^2 > (Re\ A)^2)$

lift-definition *intersects-x-axis-clmat* :: *circline-mat* ⇒ *bool* **is** *intersects-x-axis-cmat*
⟨*proof*⟩

lift-definition *intersects-x-axis* :: *circline* ⇒ *bool* **is** *intersects-x-axis-clmat*
⟨*proof*⟩

lemma *intersects-x-axis-mk-circline*:
assumes *is-real A* **and** $A \neq 0 \vee B \neq 0$
shows $intersects\text{-}x\text{-}axis\ (mk\text{-}circline\ A\ B\ (cnj\ B)\ A) \longleftrightarrow A = 0 \vee (Re\ B)^2 > (Re\ A)^2$
⟨*proof*⟩

lemma *intersects-x-axis-iff*:
assumes *is-poincare-line H*
shows $(\exists\ x \in unit\text{-}disc.\ x \in circline\text{-}set\ H \cap circline\text{-}set\ x\text{-}axis) \longleftrightarrow intersects\text{-}x\text{-}axis\ H$
⟨*proof*⟩

8.3 Check if a Poincaré line intersects the y-axis

definition *intersects-y-axis-cmat* :: *complex-mat* ⇒ *bool* **where**
[*simp*]: $intersects\text{-}y\text{-}axis\text{-}cmat\ H = (let\ (A,\ B,\ C,\ D) = H\ in\ A = 0 \vee (Im\ B)^2 > (Re\ A)^2)$

lift-definition *intersects-y-axis-clmat* :: *circline-mat* ⇒ *bool* **is** *intersects-y-axis-cmat*
⟨*proof*⟩

lift-definition *intersects-y-axis* :: *circline* ⇒ *bool* **is** *intersects-y-axis-clmat*
⟨*proof*⟩

lemma *intersects-x-axis-intersects-y-axis* [*simp*]:
shows $intersects\text{-}x\text{-}axis\ (moebius\text{-}circline\ (moebius\text{-}rotation\ (pi/2))\ H) \longleftrightarrow intersects\text{-}y\text{-}axis\ H$
⟨*proof*⟩

lemma *intersects-y-axis-iff*:
assumes *is-poincare-line H*
shows $(\exists\ y \in unit\text{-}disc.\ y \in circline\text{-}set\ H \cap circline\text{-}set\ y\text{-}axis) \longleftrightarrow intersects\text{-}y\text{-}axis\ H$ (**is** *?lhs* ⟷ *?rhs*)
⟨*proof*⟩

8.4 Intersection point of a Poincaré line with the x-axis in the unit disc

definition *calc-x-axis-intersection-cvec* :: *complex* ⇒ *complex* ⇒ *complex-vec* **where**
[*simp*]: $calc\text{-}x\text{-}axis\text{-}intersection\text{-}cvec\ A\ B =$
$(let\ discr = (Re\ B)^2 - (Re\ A)^2\ in$
$(-Re(B) + sgn\ (Re\ B) * sqrt(discr),\ A))$

definition *calc-x-axis-intersection-cmat-cvec* :: *complex-mat* ⇒ *complex-vec* **where** [*simp*]:
calc-x-axis-intersection-cmat-cvec H =
(*let* (*A*, *B*, *C*, *D*) = *H in*
if $A \neq 0$ *then*
calc-x-axis-intersection-cvec A B
else
(*0*, *1*)

)

lift-definition *calc-x-axis-intersection-clmat-hcoords* :: *circline-mat* ⇒ *complex-homo-coords* **is** *calc-x-axis-intersection-cmat-cvec*
⟨*proof*⟩

lift-definition *calc-x-axis-intersection* :: *circline* ⇒ *complex-homo* **is** *calc-x-axis-intersection-clmat-hcoords*
⟨*proof*⟩

lemma *calc-x-axis-intersection-in-unit-disc*:
assumes *is-poincare-line H intersects-x-axis H*
shows *calc-x-axis-intersection H* ∈ *unit-disc*
⟨*proof*⟩

lemma *calc-x-axis-intersection*:
assumes *is-poincare-line H* **and** *intersects-x-axis H*
shows *calc-x-axis-intersection H* ∈ *circline-set H* ∩ *circline-set x-axis*
⟨*proof*⟩

lemma *unique-calc-x-axis-intersection*:
assumes *is-poincare-line H* **and** *H* ≠ *x-axis*
assumes *x* ∈ *unit-disc* **and** *x* ∈ *circline-set H* ∩ *circline-set x-axis*
shows *x* = *calc-x-axis-intersection H*
⟨*proof*⟩

8.5 Check if an h-line intersects the positive part of the x-axis

definition *intersects-x-axis-positive-cmat* :: *complex-mat* ⇒ *bool* **where**
[*simp*]: *intersects-x-axis-positive-cmat H* = (*let* (*A*, *B*, *C*, *D*) = *H in Re A* ≠ *0* ∧ *Re B* / *Re A* < −*1*)

lift-definition *intersects-x-axis-positive-clmat* :: *circline-mat* ⇒ *bool* **is** *intersects-x-axis-positive-cmat*
⟨*proof*⟩

lift-definition *intersects-x-axis-positive* :: *circline* ⇒ *bool* **is** *intersects-x-axis-positive-clmat*
⟨*proof*⟩

lemma *intersects-x-axis-positive-mk-circline*:
assumes *is-real A* **and** *A* ≠ *0* ∨ *B* ≠ *0*
shows *intersects-x-axis-positive* (*mk-circline A B* (*cnj B*) *A*) ⟷ *Re B* / *Re A* < −*1*
⟨*proof*⟩

lemma *intersects-x-axis-positive-intersects-x-axis* [*simp*]:
assumes *intersects-x-axis-positive H*
shows *intersects-x-axis H*
⟨*proof*⟩

lemma *add-less-abs-positive-iff*:
fixes *a b* :: *real*
assumes *abs b* < *abs a*
shows *a* + *b* > *0* ⟷ *a* > *0*
⟨*proof*⟩

lemma *calc-x-axis-intersection-positive-abs′*:
fixes *A B* :: *real*
assumes $B^2 > A^2$ **and** *A* ≠ *0*
shows *abs* (*sgn*(*B*) ∗ *sqrt*($B^2 - A^2$) / *A*) < *abs*(−*B*/*A*)
⟨*proof*⟩

lemma *calc-intersect-x-axis-positive-lemma*:
assumes $B^2 > A^2$ **and** *A* ≠ *0*
shows (−*B* + *sgn B* ∗ *sqrt*($B^2 - A^2$)) / *A* > *0* ⟷ −*B*/*A* > *1*
⟨*proof*⟩

lemma *intersects-x-axis-positive-iff′*:

assumes *is-poincare-line H*
shows *intersects-x-axis-positive H* $\longleftrightarrow$
calc-x-axis-intersection H $\in$ *unit-disc* $\wedge$ *calc-x-axis-intersection H* $\in$ *circline-set H* $\cap$ *positive-x-axis* (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
⟨*proof*⟩

lemma *intersects-x-axis-positive-iff*:
assumes *is-poincare-line H* **and** *H* $\neq$ *x-axis*
shows *intersects-x-axis-positive H* $\longleftrightarrow$
($\exists$ *x. x* $\in$ *unit-disc* $\wedge$ *x* $\in$ *circline-set H* $\cap$ *positive-x-axis*) (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
⟨*proof*⟩

8.6 Check if an h-line intersects the positive part of the y-axis

definition *intersects-y-axis-positive-cmat* :: *complex-mat* $\Rightarrow$ *bool* **where**
[*simp*]: *intersects-y-axis-positive-cmat H* = (*let* (*A*, *B*, *C*, *D*) = *H in Re A* $\neq$ *0* $\wedge$ *Im B* / *Re A* < $-$*1*)

lift-definition *intersects-y-axis-positive-clmat* :: *circline-mat* $\Rightarrow$ *bool* **is** *intersects-y-axis-positive-cmat*
⟨*proof*⟩

lift-definition *intersects-y-axis-positive* :: *circline* $\Rightarrow$ *bool* **is** *intersects-y-axis-positive-clmat*
⟨*proof*⟩

lemma *intersects-x-axis-positive-intersects-y-axis-positive* [*simp*]:
shows *intersects-x-axis-positive* (*moebius-circline* (*moebius-rotation* ($-$*pi*/*2*)) *H*) $\longleftrightarrow$ *intersects-y-axis-positive H*
⟨*proof*⟩

lemma *intersects-y-axis-positive-iff*:
assumes *is-poincare-line H H* $\neq$ *y-axis*
shows ($\exists$ *y* $\in$ *unit-disc. y* $\in$ *circline-set H* $\cap$ *positive-y-axis*) $\longleftrightarrow$ *intersects-y-axis-positive H* (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
⟨*proof*⟩

8.7 Position of the intersection point in the unit disc

Check if the intersection point of one h-line with the x-axis is located more outward the edge of the disc than the intersection point of another h-line.

definition *outward-cmat* :: *complex-mat* $\Rightarrow$ *complex-mat* $\Rightarrow$ *bool* **where**
[*simp*]: *outward-cmat H1 H2* = (*let* (*A1*, *B1*, *C1*, *D1*) = *H1*; (*A2*, *B2*, *C2*, *D2*) = *H2*
in $-$*Re B1*/*Re A1* $\leq$ $-$*Re B2*/*Re A2*)
lift-definition *outward-clmat* :: *circline-mat* $\Rightarrow$ *circline-mat* $\Rightarrow$ *bool* **is** *outward-cmat*
⟨*proof*⟩
lift-definition *outward* :: *circline* $\Rightarrow$ *circline* $\Rightarrow$ *bool* **is** *outward-clmat*
⟨*proof*⟩

lemma *outward-mk-circline*:
assumes *is-real A1* **and** *is-real A2* **and** *A1* $\neq$ *0* $\vee$ *B1* $\neq$ *0* **and** *A2* $\neq$ *0* $\vee$ *B2* $\neq$ *0*
shows *outward* (*mk-circline A1 B1* (*cnj B1*) *A1*) (*mk-circline A2 B2* (*cnj B2*) *A2*) $\longleftrightarrow$ $-$ *Re B1* / *Re A1* $\leq$ $-$ *Re B2* / *Re A2*
⟨*proof*⟩

lemma *calc-x-axis-intersection-fun-mono*:
fixes *x1 x2* :: *real*
assumes *x1* > *1* **and** *x2* > *x1*
shows $x1 - sqrt(x1^2 - 1) > x2 - sqrt(x2^2 - 1)$
⟨*proof*⟩

lemma *calc-x-axis-intersection-mono*:
fixes *a1 b1 a2 b2* :: *real*
assumes $-b1/a1 > 1$ **and** $a1 \neq 0$ **and** $-b2/a2 \geq -b1/a1$ **and** $a2 \neq 0$
shows $(-b1 + sgn\ b1 * sqrt(b1^2 - a1^2)) / a1 \geq (-b2 + sgn\ b2 * sqrt(b2^2 - a2^2)) / a2$ (**is** *?lhs* $\geq$ *?rhs*)
⟨*proof*⟩

lemma *outward*:
assumes *is-poincare-line H1* **and** *is-poincare-line H2*
assumes *intersects-x-axis-positive H1* **and** *intersects-x-axis-positive H2*

assumes *outward H1 H2*
shows *Re* (*to-complex* (*calc-x-axis-intersection H1*)) ≥ *Re* (*to-complex* (*calc-x-axis-intersection H2*))
⟨*proof*⟩

8.8 Ideal points and x-axis intersection

lemma *ideal-points-intersects-x-axis*:
assumes *is-poincare-line H* **and** *ideal-points H* = {*i1*, *i2*} **and** *H* ≠ *x-axis*
shows *intersects-x-axis H* ⟷ *Im* (*to-complex i1*) ∗ *Im* (*to-complex i2*) < *0*
⟨*proof*⟩

end
theory *Poincare-Perpendicular*
imports *Poincare-Lines-Axis-Intersections*
begin

9 H-perpendicular h-lines in the Poincaré model

definition *perpendicular-to-x-axis-cmat* :: *complex-mat* ⇒ *bool* **where**
[*simp*]: *perpendicular-to-x-axis-cmat H* ⟷ (*let* (*A*, *B*, *C*, *D*) = *H in is-real B*)

lift-definition *perpendicular-to-x-axis-clmat* :: *circline-mat* ⇒ *bool* **is** *perpendicular-to-x-axis-cmat*
⟨*proof*⟩

lift-definition *perpendicular-to-x-axis* :: *circline* ⇒ *bool* **is** *perpendicular-to-x-axis-clmat*
⟨*proof*⟩

lemma *perpendicular-to-x-axis*:
assumes *is-poincare-line H*
shows *perpendicular-to-x-axis H* ⟷ *perpendicular x-axis H*
⟨*proof*⟩

lemma *perpendicular-to-x-axis-y-axis*:
assumes *perpendicular-to-x-axis* (*poincare-line* 0_h (*of-complex z*)) *z* ≠ *0*
shows *is-imag z*
⟨*proof*⟩

lemma *wlog-perpendicular-axes*:
assumes *in-disc*: *u* ∈ *unit-disc v* ∈ *unit-disc z* ∈ *unit-disc*
assumes *perpendicular*: *is-poincare-line H1 is-poincare-line H2 perpendicular H1 H2*
assumes *z* ∈ *circline-set H1* ∩ *circline-set H2 u* ∈ *circline-set H1 v* ∈ *circline-set H2*
assumes *axes*: ⋀ *x y*. ⟦*is-real x*; *0* ≤ *Re x*; *Re x* < *1*; *is-imag y*; *0* ≤ *Im y*; *Im y* < *1*⟧ ⟹ *P* 0_h (*of-complex x*) (*of-complex y*)
assumes *moebius*: ⋀ *M u v w*. ⟦*unit-disc-fix M*; *u* ∈ *unit-disc*; *v* ∈ *unit-disc*; *w* ∈ *unit-disc*; *P* (*moebius-pt M u*) (*moebius-pt M v*) (*moebius-pt M w*) ⟧ ⟹ *P u v w*
assumes *conjugate*: ⋀ *u v w*. ⟦*u* ∈ *unit-disc*; *v* ∈ *unit-disc*; *w* ∈ *unit-disc*; *P* (*conjugate u*) (*conjugate v*) (*conjugate w*) ⟧ ⟹ *P u v w*
shows *P z u v*
⟨*proof*⟩

lemma *wlog-perpendicular-foot*:
assumes *in-disc*: *u* ∈ *unit-disc v* ∈ *unit-disc w* ∈ *unit-disc z* ∈ *unit-disc*
assumes *perpendicular*: *u* ≠ *v is-poincare-line H perpendicular* (*poincare-line u v*) *H*
assumes *z* ∈ *circline-set* (*poincare-line u v*) ∩ *circline-set H w* ∈ *circline-set H*
assumes *axes*: ⋀ *u v w*. ⟦*is-real u*; *0* < *Re u*; *Re u* < *1*; *is-real v*; −*1* < *Re v*; *Re v* < *1*; *Re u* ≠ *Re v*; *is-imag w*; *0* ≤ *Im w*; *Im w* < *1*⟧ ⟹ *P* 0_h (*of-complex u*) (*of-complex v*) (*of-complex w*)
assumes *moebius*: ⋀ *M z u v w*. ⟦*unit-disc-fix M*; *u* ∈ *unit-disc*; *v* ∈ *unit-disc*; *w* ∈ *unit-disc*; *z* ∈ *unit-disc*; *P* (*moebius-pt M z*) (*moebius-pt M u*) (*moebius-pt M v*) (*moebius-pt M w*) ⟧ ⟹ *P z u v w*
assumes *conjugate*: ⋀ *z u v w*. ⟦*u* ∈ *unit-disc*; *v* ∈ *unit-disc*; *w* ∈ *unit-disc*; *P* (*conjugate z*) (*conjugate u*) (*conjugate v*) (*conjugate w*) ⟧ ⟹ *P z u v w*
assumes *perm*: *P z v u w* ⟹ *P z u v w*
shows *P z u v w*
⟨*proof*⟩

lemma *perpendicular-to-x-axis-intersects-x-axis*:

assumes *is-poincare-line H perpendicular-to-x-axis H*
shows *intersects-x-axis H*
⟨*proof*⟩

lemma *perpendicular-intersects*:
assumes *is-poincare-line H1 is-poincare-line H2*
assumes *perpendicular H1 H2*
shows ∃ *z. z* ∈ *unit-disc* ∧ *z* ∈ *circline-set H1* ∩ *circline-set H2* (**is** *?P′ H1 H2*)
⟨*proof*⟩

definition *calc-perpendicular-to-x-axis-cmat* :: *complex-vec* ⇒ *complex-mat* **where**
[*simp*]: *calc-perpendicular-to-x-axis-cmat z* =
(*let* (*z1*, *z2*) = *z*
in if z1∗*cnj z2* + *z2*∗*cnj z1* = *0 then*
(*0*, *1*, *1*, *0*)
else
let A = *z1*∗*cnj z2* + *z2*∗*cnj z1*;
B = −(*z1*∗*cnj z1* + *z2*∗*cnj z2*)
in (*A*, *B*, *B*, *A*)
)

lift-definition *calc-perpendicular-to-x-axis-clmat* :: *complex-homo-coords* ⇒ *circline-mat* **is** *calc-perpendicular-to-x-axis-cmat*
⟨*proof*⟩

lift-definition *calc-perpendicular-to-x-axis* :: *complex-homo* ⇒ *circline* **is** *calc-perpendicular-to-x-axis-clmat*
⟨*proof*⟩

lemma *calc-perpendicular-to-x-axis*:
assumes *z* ≠ *of-complex 1 z* ≠ *of-complex* (−*1*)
shows *z* ∈ *circline-set* (*calc-perpendicular-to-x-axis z*) ∧
is-poincare-line (*calc-perpendicular-to-x-axis z*) ∧
perpendicular-to-x-axis (*calc-perpendicular-to-x-axis z*)
⟨*proof*⟩

lemma *ex-perpendicular*:
assumes *is-poincare-line H z* ∈ *unit-disc*
shows ∃ *H′*. *is-poincare-line H′* ∧ *perpendicular H H′* ∧ *z* ∈ *circline-set H′* (**is** *?P′ H z*)
⟨*proof*⟩

lemma *ex-perpendicular-foot*:
assumes *is-poincare-line H z* ∈ *unit-disc*
shows ∃ *H′*. *is-poincare-line H′* ∧ *z* ∈ *circline-set H′* ∧ *perpendicular H H′* ∧
(∃ *z′* ∈ *unit-disc*. *z′* ∈ *circline-set H′* ∩ *circline-set H*)
⟨*proof*⟩

lemma *Pythagoras*:
assumes *in-disc*: *u* ∈ *unit-disc v* ∈ *unit-disc w* ∈ *unit-disc v* ≠ *w*
assumes *distinct*[*u*, *v*, *w*] ⟶ *perpendicular* (*poincare-line u v*) (*poincare-line u w*)
shows *cosh* (*poincare-distance v w*) = *cosh* (*poincare-distance u v*) ∗ *cosh* (*poincare-distance u w*) (**is** *?P′ u v w*)
⟨*proof*⟩

end

10 Poincaré disc model types

In this section we introduce datatypes that represent objects in the Poincaré disc model. The types are defined as subtypes (e.g., the h-points are defined as elements of $\mathbb{C}P^1$ that lie within the unit disc). The functions on those types are defined by lifting the functions defined on the carrier type (e.g., h-distance is defined by lifting the distance function defined for elements of $\mathbb{C}P^1$).

theory *Poincare*
imports *Poincare-Lines Poincare-Between Poincare-Distance Poincare-Circles*
begin

10.1 H-points

typedef *p-point* = {*z*. *z* ∈ *unit-disc*}
⟨*proof*⟩

setup-lifting *type-definition-p-point*

Point zero

lift-definition *p-zero* :: *p-point* **is** 0_h
⟨*proof*⟩

Constructing h-points from complex numbers

lift-definition *p-of-complex* :: *complex* ⇒ *p-point* **is** λ *z*. *if cmod z* < *1 then of-complex z else* 0_h
⟨*proof*⟩

10.2 H-lines

typedef *p-line* = {*H*. *is-poincare-line H*}
⟨*proof*⟩

setup-lifting *type-definition-p-line*

lift-definition *p-incident* :: *p-line* ⇒ *p-point* ⇒ *bool* **is** *on-circline*
⟨*proof*⟩

Set of h-points on an h-line

definition *p-points* :: *p-line* ⇒ *p-point set* **where**
p-points l = {*p*. *p-incident l p*}

x-axis is an example of an h-line

lift-definition *p-x-axis* :: *p-line* **is** *x-axis*
⟨*proof*⟩

Constructing the unique h-line from two h-points

lift-definition *p-line* :: *p-point* ⇒ *p-point* ⇒ *p-line* **is** *poincare-line*
⟨*proof*⟩

Next we show how to lift some lemmas. This could be done for all the lemmas that we have proved earlier, but we do not do that.

If points are different then the constructed line contains the starting points

lemma *p-on-line*:
assumes *z* ≠ *w*
shows *p-incident* (*p-line z w*) *z*
p-incident (*p-line z w*) *w*
⟨*proof*⟩

There is a unique h-line passing trough the two different given h-points

lemma
assumes *u* ≠ *v*
shows ∃! *l*. {*u*, *v*} ⊆ *p-points l*
⟨*proof*⟩

The unique h-line trough zero and a non-zero h-point on the x-axis is the x-axis

lemma
assumes *p-zero* ∈ *p-points l u* ∈ *p-points l u* ≠ *p-zero u* ∈ *p-points p-x-axis*
shows *l* = *p-x-axis*
⟨*proof*⟩

10.3 H-collinearity

lift-definition *p-collinear* :: *p-point set* ⇒ *bool* **is** *poincare-collinear*
⟨*proof*⟩

10.4 H-isometries

H-isometries are functions that map the unit disc onto itself

typedef *p-isometry* = {*f*. *unit-disc-fix-f f*}
⟨*proof*⟩

setup-lifting *type-definition-p-isometry*

Action of an h-isometry on an h-point

lift-definition *p-isometry-pt* :: *p-isometry* ⇒ *p-point* ⇒ *p-point* **is** λ *f p*. *f p*
⟨*proof*⟩

Action of an h-isometry on an h-line

lift-definition *p-isometry-line* :: *p-isometry* ⇒ *p-line* ⇒ *p-line* **is** λ *f l*. *unit-disc-fix-f-circline f l*
⟨*proof*⟩

An example lemma about h-isometries.

H-isometries preserve h-collinearity

lemma *p-collinear-p-isometry-pt* [*simp*]:
shows *p-collinear* (*p-isometry-pt M* ‘ *A*) ⟷ *p-collinear A*
⟨*proof*⟩

10.5 H-distance and h-congruence

lift-definition *p-dist* :: *p-point* ⇒ *p-point* ⇒ *real* **is** *poincare-distance*
⟨*proof*⟩

definition *p-congruent* :: *p-point* ⇒ *p-point* ⇒ *p-point* ⇒ *p-point* ⇒ *bool* **where**
[*simp*]: *p-congruent u v u′ v′* ⟷ *p-dist u v* = *p-dist u′ v′*

lemma
assumes *p-dist u v* = *p-dist u′ v′*
assumes *p-dist v w* = *p-dist v′ w′*
assumes *p-dist u w* = *p-dist u′ w′*
shows ∃ *f*. *p-isometry-pt f u* = *u′* ∧ *p-isometry-pt f v* = *v′* ∧ *p-isometry-pt f w* = *w′*
⟨*proof*⟩

We prove that unit disc equipped with Poincaré distance is a metric space, i.e. an instantiation of *metric-space* locale.

instantiation *p-point* :: *metric-space*
begin
definition *dist-p-point* = *p-dist*
definition (*uniformity-p-point* :: (*p-point* × *p-point*) *filter*) = (*INF e*∈{*0*<..}. *principal* {(*x*, *y*). *dist-class.dist x y* < *e*})
definition *open-p-point* (*U* :: *p-point set*) = (∀ *x* ∈ *U*. *eventually* (λ(*x′*, *y*). *x′* = *x* ⟶ *y* ∈ *U*) *uniformity*)
instance
⟨*proof*⟩
end

10.6 H-betweennes

lift-definition *p-between* :: *p-point* ⇒ *p-point* ⇒ *p-point* ⇒ *bool* **is** *poincare-between*
⟨*proof*⟩

end

11 Poincaré model satisfies Tarski axioms

theory *Poincare-Tarski*
imports *Poincare Poincare-Lines-Axis-Intersections Tarski*
begin

11.1 Pasch axiom

lemma *Pasch-fun-mono*:
fixes *r1 r2* :: *real*
assumes $0 < r1$ **and** $r1 \leq r2$ **and** $r2 < 1$
shows $r1 + 1/r1 \geq r2 + 1/r2$
⟨*proof*⟩

Pasch axiom, non-degenerative case.

lemma *Pasch-nondeg*:
assumes $x \in$ *unit-disc* **and** $y \in$ *unit-disc* **and** $z \in$ *unit-disc* **and** $u \in$ *unit-disc* **and** $v \in$ *unit-disc*
assumes *distinct* $[x, y, z, u, v]$
assumes ¬ *poincare-collinear* $\{x, y, z\}$
assumes *poincare-between x u z* **and** *poincare-between y v z*
shows ∃ *a*. $a \in$ *unit-disc* ∧ *poincare-between u a y* ∧ *poincare-between x a v*
⟨*proof*⟩

Pasch axiom, only degenerative cases.

lemma *Pasch-deg*:
assumes $x \in$ *unit-disc* **and** $y \in$ *unit-disc* **and** $z \in$ *unit-disc* **and** $u \in$ *unit-disc* **and** $v \in$ *unit-disc*
assumes ¬ *distinct* $[x, y, z, u, v]$ ∨ *poincare-collinear* $\{x, y, z\}$
assumes *poincare-between x u z* **and** *poincare-between y v z*
shows ∃ *a*. $a \in$ *unit-disc* ∧ *poincare-between u a y* ∧ *poincare-between x a v*
⟨*proof*⟩

Axiom of Pasch

lemma *Pasch*:
assumes $x \in$ *unit-disc* **and** $y \in$ *unit-disc* **and** $z \in$ *unit-disc* **and** $u \in$ *unit-disc* **and** $v \in$ *unit-disc*
assumes *poincare-between x u z* **and** *poincare-between y v z*
shows ∃ *a*. $a \in$ *unit-disc* ∧ *poincare-between u a y* ∧ *poincare-between x a v*
⟨*proof*⟩

11.2 Segment construction axiom

lemma *segment-construction*:
assumes $x \in$ *unit-disc* **and** $y \in$ *unit-disc*
assumes $a \in$ *unit-disc* **and** $b \in$ *unit-disc*
shows ∃ *z*. $z \in$ *unit-disc* ∧ *poincare-between x y z* ∧ *poincare-distance y z* = *poincare-distance a b*
⟨*proof*⟩

11.3 Five segment axiom

lemma *five-segment-axiom*:
assumes
in-disc: $x \in$ *unit-disc* $y \in$ *unit-disc* $z \in$ *unit-disc* $u \in$ *unit-disc* **and**
in-disc': $x' \in$ *unit-disc* $y' \in$ *unit-disc* $z' \in$ *unit-disc* $u' \in$ *unit-disc* **and**
$x \neq y$ **and**
betw: *poincare-between x y z poincare-between x' y' z'* **and**
xy: *poincare-distance x y* = *poincare-distance x' y'* **and**
xu: *poincare-distance x u* = *poincare-distance x' u'* **and**
yu: *poincare-distance y u* = *poincare-distance y' u'* **and**
yz: *poincare-distance y z* = *poincare-distance y' z'*
shows
poincare-distance z u = *poincare-distance z' u'*
⟨*proof*⟩

11.4 Upper dimension axiom

lemma *upper-dimension-axiom*:
assumes *in-disc*: $x \in$ *unit-disc* $y \in$ *unit-disc* $z \in$ *unit-disc* $u \in$ *unit-disc* $v \in$ *unit-disc*
assumes *poincare-distance x u* = *poincare-distance x v*
poincare-distance y u = *poincare-distance y v*
poincare-distance z u = *poincare-distance z v*
$u \neq v$
shows *poincare-between x y z* ∨ *poincare-between y z x* ∨ *poincare-between z x y*
⟨*proof*⟩

11.5 Lower dimension axiom

lemma *lower-dimension-axiom*:
  shows $\exists\ a \in$ *unit-disc*. $\exists\ b \in$ *unit-disc*. $\exists\ c \in$ *unit-disc*.
    $\neg$ *poincare-between a b c* $\wedge$ $\neg$ *poincare-between b c a* $\wedge$ $\neg$ *poincare-between c a b*
⟨*proof*⟩

11.6 Negated Euclidean axiom

lemma *negated-euclidean-axiom-aux*:
  assumes *on-circline H* (*of-complex* $(1/2 + i/2)$) **and** *is-poincare-line H*
  assumes *intersects-x-axis-positive H*
  shows $\neg$ *intersects-y-axis-positive H*
  ⟨*proof*⟩

lemma *negated-euclidean-axiom*:
  shows $\exists\ a\ b\ c\ d\ t$.
    $a \in$ *unit-disc* $\wedge$ $b \in$ *unit-disc* $\wedge$ $c \in$ *unit-disc* $\wedge$ $d \in$ *unit-disc* $\wedge$ $t \in$ *unit-disc* $\wedge$
    poincare-between a d t $\wedge$ *poincare-between b d c* $\wedge$ $a \neq d$ $\wedge$
      ($\forall\ x\ y.\ x \in$ *unit-disc* $\wedge$ $y \in$ *unit-disc* $\wedge$
        poincare-between a b x $\wedge$ *poincare-between x t y* $\longrightarrow$ $\neg$ *poincare-between a c y*)
⟨*proof*⟩

Alternate form of the Euclidean axiom – this one is much easier to prove

lemma *negated-euclidean-axiom′*:
  shows $\exists\ a\ b\ c$.
    $a \in$ *unit-disc* $\wedge$ $b \in$ *unit-disc* $\wedge$ $c \in$ *unit-disc* $\wedge$ $\neg$(*poincare-collinear* $\{a, b, c\}$) $\wedge$
      $\neg$($\exists\ x.\ x \in$ *unit-disc* $\wedge$
        poincare-distance a x = *poincare-distance b x* $\wedge$
        poincare-distance a x = *poincare-distance c x*)
⟨*proof*⟩

11.7 Continuity axiom

The set ϕ is on the left of the set ψ

abbreviation *set-order* **where**
 set-order A $\varphi\ \psi \equiv \forall x \in$ *unit-disc*. $\forall y \in$ *unit-disc*. $\varphi\ x \wedge \psi\ y \longrightarrow$ *poincare-between A x y*

The point B is between the sets ϕ and ψ

abbreviation *point-between-sets* **where**
 point-between-sets φ *B* $\psi \equiv \forall x \in$ *unit-disc*. $\forall y \in$ *unit-disc*. $\varphi\ x \wedge \psi\ y \longrightarrow$ *poincare-between x B y*

lemma *continuity*:
  assumes $\exists\ A \in$ *unit-disc*. *set-order A* $\varphi\ \psi$
  shows $\exists\ B \in$ *unit-disc*. *point-between-sets* φ *B* ψ
⟨*proof*⟩

11.8 Limiting parallels axiom

Auxiliary definitions

definition *poincare-on-line* **where**
  poincare-on-line p a b $\longleftrightarrow$ *poincare-collinear* $\{p, a, b\}$

definition *poincare-on-ray* **where**
  poincare-on-ray p a b $\longleftrightarrow$ *poincare-between a p b* $\vee$ *poincare-between a b p*

definition *poincare-in-angle* **where**
  poincare-in-angle p a b c $\longleftrightarrow$
    $b \neq a \wedge b \neq c \wedge p \neq b \wedge (\exists\ x \in$ *unit-disc*. *poincare-between a x c* $\wedge\ x \neq a \wedge x \neq c \wedge$ *poincare-on-ray p b x*)

definition *poincare-ray-meets-line* **where**
  poincare-ray-meets-line a b c d $\longleftrightarrow$ ($\exists\ x \in$ *unit-disc*. *poincare-on-ray x a b* $\wedge$ *poincare-on-line x c d*)

All points on ray are collinear

lemma *poincare-on-ray-poincare-collinear*:

assumes *p* ∈ *unit-disc* **and** *a* ∈ *unit-disc* **and** *b* ∈ *unit-disc* **and** *poincare-on-ray p a b*
shows *poincare-collinear* {*p*, *a*, *b*}
⟨*proof*⟩

H-isometries preserve all defined auxiliary relations

lemma *unit-disc-fix-preserves-poincare-on-line* [*simp*]:
assumes *unit-disc-fix M* **and** *p* ∈ *unit-disc a* ∈ *unit-disc b* ∈ *unit-disc*
shows *poincare-on-line* (*moebius-pt M p*) (*moebius-pt M a*) (*moebius-pt M b*) ⟷ *poincare-on-line p a b*
⟨*proof*⟩

lemma *unit-disc-fix-preserves-poincare-on-ray* [*simp*]:
assumes *unit-disc-fix M p* ∈ *unit-disc a* ∈ *unit-disc b* ∈ *unit-disc*
shows *poincare-on-ray* (*moebius-pt M p*) (*moebius-pt M a*) (*moebius-pt M b*) ⟷ *poincare-on-ray p a b*
⟨*proof*⟩

lemma *unit-disc-fix-preserves-poincare-in-angle* [*simp*]:
assumes *unit-disc-fix M p* ∈ *unit-disc a* ∈ *unit-disc b* ∈ *unit-disc c* ∈ *unit-disc*
shows *poincare-in-angle* (*moebius-pt M p*) (*moebius-pt M a*) (*moebius-pt M b*) (*moebius-pt M c*) ⟷ *poincare-in-angle*
p a b c (**is** *?lhs* ⟷ *?rhs*)
⟨*proof*⟩

lemma *unit-disc-fix-preserves-poincare-ray-meets-line* [*simp*]:
assumes *unit-disc-fix M a* ∈ *unit-disc b* ∈ *unit-disc c* ∈ *unit-disc d* ∈ *unit-disc*
shows *poincare-ray-meets-line* (*moebius-pt M a*) (*moebius-pt M b*) (*moebius-pt M c*) (*moebius-pt M d*) ⟷ *poincare-ray-meets-line*
a b c d (**is** *?lhs* ⟷ *?rhs*)
⟨*proof*⟩

H-lines that intersect on the absolute do not meet (they do not share a common h-point)

lemma *tangent-not-meet*:
assumes *x1* ∈ *unit-disc* **and** *x2* ∈ *unit-disc* **and** *x1* ≠ *x2* **and** ¬ *poincare-collinear* {0_h, *x1*, *x2*}
assumes *i* ∈ *ideal-points* (*poincare-line x1 x2*) *a* ∈ *unit-disc a* ≠ 0_h *poincare-collinear* {0_h, *a*, *i*}
shows ¬ *poincare-ray-meets-line* 0_h *a x1 x2*
⟨*proof*⟩

lemma *limiting-parallels*:
assumes *a* ∈ *unit-disc* **and** *x1* ∈ *unit-disc* **and** *x2* ∈ *unit-disc* **and** ¬ *poincare-on-line a x1 x2*
shows ∃ *a1*∈*unit-disc*. ∃ *a2*∈*unit-disc*.
¬ *poincare-on-line a a1 a2* ∧
¬ *poincare-ray-meets-line a a1 x1 x2* ∧ ¬ *poincare-ray-meets-line a a2 x1 x2* ∧
(∀ *a′*∈*unit-disc*. *poincare-in-angle a′ a1 a a2* ⟶ *poincare-ray-meets-line a a′ x1 x2*) (**is** *?P a x1 x2*)
⟨*proof*⟩

11.9 Interpretation of locales

global-interpretation *PoincareTarskiAbsolute*: *TarskiAbsolute* **where** *cong* = *p-congruent* **and** *betw* = *p-between*
defines *p-on-line* = *PoincareTarskiAbsolute.on-line* **and**
p-on-ray = *PoincareTarskiAbsolute.on-ray* **and**
p-in-angle = *PoincareTarskiAbsolute.in-angle* **and**
p-ray-meets-line = *PoincareTarskiAbsolute.ray-meets-line*
⟨*proof*⟩

interpretation *PoincareTarskiHyperbolic*: *TarskiHyperbolic*
where *cong* = *p-congruent* **and** *betw* = *p-between*
⟨*proof*⟩

interpretation *PoincareElementaryTarskiHyperbolic*: *ElementaryTarskiHyperbolic p-congruent p-between*
⟨*proof*⟩

end

References

[1] R. Coghetto. Klein-Beltrami Model. Part I. *Formalized Mathematics*, 26(1):21–32, 2018.

[2] R. Coghetto. Klein-Beltrami Model. Part II. *Formalized Mathematics*, 26(1):33–48, 2018.

[3] N. Lobatschewsky. *Geometrische Untersuchungen zur Theorie der Parallellinien*, pages 159–223. Springer Vienna, Vienna, 1985.

[4] T. J. M. Makarios. A Mechanical Verification of the Independence of Tarski's Euclidean Axiom. Master's thesis, Victoria University of Wellington, 2012. Master Thesis.

[5] F. Marić and D. Simić. Formalizing Complex Plane Geometry. *Annals of Mathematics and Artificial Intelligence*, 74(3-4):271–308, 2015.

[6] F. Mari and D. Simi. Complex geometry. *Archive of Formal Proofs*, Dec. 2019. http://isa-afp.org/entries/Complex_Geometry.html, Formal proof development.

[7] W. Schwabhäuser, W. Szmielew, and A. Tarski. *Metamathematische Methoden in der Geometrie*. Springer-Verlag, Berlin, 1983.

[8] H. Schwerdtfeger. *Geometry of Complex Numbers: Circle Geometry, Moebius Transformation, Non-euclidean Geometry*. Courier Corporation, 1979.