

Moebius

By filip

January 3, 2014

Contents

1	More about complex numbers	2
2	Systems of linear equations	16
3	Quadratic equations	17
4	Vectors, Matrices	18
4.1	Vectors	18
4.2	Matrices	19
4.3	Eigenvalues and eigenvectors	25
4.4	Bilinear and Quadratic forms; Congruence	25
5	Unitary matrices	27
6	Hermitean matrices	33
7	Elementary complex geometry	34
8	Homogeneous coordinates in extended complex plane	41
8.1	Ratio and crossratio	50
8.2	Distance	52
9	Riemann sphere	55
10	Moebius transformations	60
10.1	Action on points	61
10.2	Moebius group	62
10.3	Special kinds of Moebius transformations	64
10.4	Decomposition	67
10.5	Cross ratio and moebius existence	68
10.6	Fixed points and moebius uniqueness	68
10.7	Pole	69
10.8	Antihomographies	70

10.9 Classification	70
11 Circline	72
11.1 Circline definition	72
11.2 Connections with circles on the Riemann sphere	75
11.3 Some special circlines	76
11.4 Moebius action on circlines	78
11.5 Conjugation, reciprocation and inversion of circlines	79
11.6 Circline uniqueness	80
11.6.1 Zero type circline uniqueness	80
11.6.2 Negative type circline uniqueness	80
11.7 Circline set cardinality	81
11.7.1 Diagonal circlines	81
11.7.2 Zero type circline set cardinality	81
11.7.3 Negative type circline set cardinality	82
11.7.4 Positive type circline set cardinality	82
11.7.5 Cardinality determines type	82
11.7.6 Circline set is injective	82
11.8 Symmetric points wrt. circline	82
11.9 Oriented circlines; discs	83
11.10 Some special oriented circlines and discs	87
11.11 Moebius action on oriented circlines and discs	88
11.12 Oriented circlines uniqueness	90
11.13 Disc automorphisms	92
11.14 Angle between circlines	95

1 More about complex numbers

```
theory MoreComplex
imports Complex-Main
begin
```

```
lemma mult-pow2-lt0:
  assumes  $b \neq 0$ 
  shows  $a < 0 \longleftrightarrow b^2 * a < (0::real)$ 
<proof>
```

```
lemma mult-pow2-gt0:
  assumes  $b \neq 0$ 
  shows  $a > 0 \longleftrightarrow b^2 * a > (0::real)$ 
<proof>
```

```
lemma square-cancel:
  assumes  $a^2 \geq b^2$   $a \geq 0$   $b \geq 0$ 
  shows  $a \geq b$ 
```

shows $a \geq b$
 $\langle proof \rangle$

lemmas $complex-cnj = complex-cnj-diff\ complex-cnj-mult\ complex-cnj-add\ complex-cnj-divide$
 $complex-cnj-minus$

abbreviation $cor \equiv complex-of-real$

lemma $[simp]: cor\ -1 = -1$
 $\langle proof \rangle$

lemma $[simp]: -\ cor\ -1 = 1$
 $\langle proof \rangle$

lemma $rcis-cnj: cnj\ a = rcis\ (cmod\ a)\ (-\ arg\ a)$
 $\langle proof \rangle$

lemma $cmod-cis\ [simp]:$
assumes $a \neq 0$
shows $cor\ (cmod\ a) * cis\ (arg\ a) = a$
 $\langle proof \rangle$

lemma $cis-cmod\ [simp]:$
assumes $a \neq 0$
shows $cis\ (arg\ a) * cor\ (cmod\ a) = a$
 $\langle proof \rangle$

lemma $cor-squared: (cor\ x)^2 = cor\ (x^2)$
 $\langle proof \rangle$

lemma $cor-add: cor\ (a + b) = cor\ a + cor\ b$
 $\langle proof \rangle$

lemma $cor-mult: cor\ (a * b) = cor\ a * cor\ b$
 $\langle proof \rangle$

lemma $cor-sqrt-mult-cor-sqrt\ [simp]:$
shows $cor\ (sqrt\ A) * cor\ (sqrt\ A) = cor\ |A|$
 $\langle proof \rangle$

lemma $[simp]: (Complex\ a\ b) * 2 = Complex\ (2*a)\ (2*b)$
 $\langle proof \rangle$

lemma $re-complex:$
 $Complex\ (Re\ z)\ 0 = (z + cnj\ z)/2$

$\langle proof \rangle$

lemma *im-complex*:

Complex 0 (Im z) = (z - cnj z)/2

$\langle proof \rangle$

lemma *Complex-scale1*: *Complex (a * b) (a * c) = cor a * Complex b c*

$\langle proof \rangle$

lemma *Complex-scale2*: *Complex (a * c) (b * c) = Complex a b * cor c*

$\langle proof \rangle$

lemma *Complex-scale3*: *Complex (a / b) (a / c) = cor a * Complex (1 / b) (1 / c)*

$\langle proof \rangle$

lemma *Complex-scale4*: *c ≠ 0 ⇒ Complex (a / c) (b / c) = Complex a b / cor c*

$\langle proof \rangle$

lemma *complex-mult-cnj-cmod*:

*z * cnj z = cor ((cmod z)²)*

$\langle proof \rangle$

lemma

*cmod-square: (cmod z)² = Re (z * cnj z)*

$\langle proof \rangle$

lemma *cnjE*:

assumes *x ≠ 0*

shows *cnj x = cor ((cmod x)²) / x*

$\langle proof \rangle$

lemma *cmod-mult [simp]*: *cmod (a * b) = cmod a * cmod b*

$\langle proof \rangle$

lemma *cmod-divide [simp]*: *cmod (a / b) = cmod a / cmod b*

$\langle proof \rangle$

lemma *[simp]*: *cmod (z / cor k) = cmod z / |k|*

$\langle proof \rangle$

lemma *[simp]*: *cmod (z*z1 - z*z2) = cmod z*cmod(z1 - z2)*

$\langle proof \rangle$

lemma *cmod-eqI*:

assumes *z1 * cnj z1 = z2 * cnj z2*

shows $\text{cmod } z1 = \text{cmod } z2$
 $\langle \text{proof} \rangle$

lemma *cmod-eqE*:
assumes $\text{cmod } z1 = \text{cmod } z2$
shows $z1 * \text{cnj } z1 = z2 * \text{cnj } z2$
 $\langle \text{proof} \rangle$

lemma [*simp*]: $\text{cmod } a = 1 \longleftrightarrow a * \text{cnj } a = 1$
 $\langle \text{proof} \rangle$

abbreviation *is-real* **where**
 $\text{is-real } z \equiv \text{Im } z = 0$

lemma *complex-eq-if-Re-eq*:
assumes $\text{is-real } z1 \text{ is-real } z2$
shows $z1 = z2 \longleftrightarrow \text{Re } z1 = \text{Re } z2$
 $\langle \text{proof} \rangle$

lemma *mult-reals*:
assumes $\text{is-real } a \text{ is-real } b$
shows $\text{is-real } (a * b)$
 $\langle \text{proof} \rangle$

lemma *div-reals*:
assumes $\text{is-real } a \text{ is-real } b$
shows $\text{is-real } (a / b)$
 $\langle \text{proof} \rangle$

lemma *complex-of-real-Re*:
assumes $\text{is-real } k$
shows $\text{cor } (\text{Re } k) = k$
 $\langle \text{proof} \rangle$

lemma *is-real-complex-of-real*:
 $\text{is-real } (\text{cor } x)$
 $\langle \text{proof} \rangle$

lemma *cor-cmod-real*:
assumes $\text{is-real } a$
shows $\text{cor } (\text{cmod } a) = a \vee \text{cor } (\text{cmod } a) = -a$
 $\langle \text{proof} \rangle$

lemma *eq-cnj-iff-real*:
 $z = \text{cnj } z \longleftrightarrow \text{is-real } z$
 $\langle \text{proof} \rangle$

lemma *Re-divide-real*:
assumes *is-real* b $b \neq 0$
shows $\text{Re } (a / b) = (\text{Re } a) / (\text{Re } b)$
 $\langle \text{proof} \rangle$

lemma *Re-mult-real*:
assumes *is-real* a
shows $\text{Re } (a * b) = (\text{Re } a) * (\text{Re } b)$
 $\langle \text{proof} \rangle$

lemma *Im-mult-real*:
assumes *is-real* a
shows $\text{Im } (a * b) = (\text{Re } a) * (\text{Im } b)$
 $\langle \text{proof} \rangle$

lemma *Im-divide-real*:
assumes *is-real* b $b \neq 0$
shows $\text{Im } (a / b) = (\text{Im } a) / (\text{Re } b)$
 $\langle \text{proof} \rangle$

lemma [*simp*]: $\text{Re } (x / 2) = \text{Re } x / 2$
 $\langle \text{proof} \rangle$

lemma [*simp*]: $\text{Re } (2 * x) = 2 * \text{Re } x$
 $\langle \text{proof} \rangle$

lemma *Re-sgn*:
assumes *is-real* R
shows $\text{Re } (\text{sgn } R) = \text{sgn } (\text{Re } R)$
 $\langle \text{proof} \rangle$

abbreviation *rot90* **where**
 $\text{rot90 } z \equiv \text{Complex } (-\text{Im } z) (\text{Re } z)$

lemma *rot90-ii*: $\text{rot90 } z = z * ii$
 $\langle \text{proof} \rangle$

abbreviation *cnj-mix* **where**
 $\text{cnj-mix } z1 \ z2 \equiv \text{cnj } z1 * z2 + z1 * \text{cnj } z2$

lemma *cnj-mix-minus*:
shows $\text{cnj } z1 * z2 - z1 * \text{cnj } z2 = ii * \text{cnj-mix } (\text{rot90 } z1) \ z2$
 $\langle \text{proof} \rangle$

lemma *cnj-mix-minus'*:
shows $\text{cnj } z1 * z2 - z1 * \text{cnj } z2 = \text{rot90 } (\text{cnj-mix } (\text{rot90 } z1) \ z2)$

$\langle proof \rangle$

lemma *cnj-mix-real*:
 is-real (cnj-mix z1 z2)
 $\langle proof \rangle$

abbreviation *scalprod* **where**
 scalprod z1 z2 \equiv *cnj-mix z1 z2 / 2*

lemma *cos-periodic-pi2*: *cos (pi + x) = - cos x*
 $\langle proof \rangle$

lemma *cos-periodic-pi3*: *cos (x - pi) = - cos x*
 $\langle proof \rangle$

lemma *cos-periodic-4 [simp]*: *cos (pi - x) = - cos x*
 $\langle proof \rangle$

lemma *sin-periodic-pi3*: *sin (x - pi) = - sin x*
 $\langle proof \rangle$

lemma *cos-lt-zero*:
 assumes *x > pi/2 x ≤ pi*
 shows *cos x < 0*
 $\langle proof \rangle$

lemma *sin-kpi*:
 fixes k::int
 shows *sin (real k * pi) = 0*
 $\langle proof \rangle$

lemma *cos-kpi-odd*:
 fixes k::int
 assumes *odd k*
 shows *cos (real k * pi) = -1*
 $\langle proof \rangle$

lemma *cos-kpi-even*:
 fixes k::int
 assumes *even k*
 shows *cos (real k * pi) = 1*
 $\langle proof \rangle$

lemma *sin-pi2-kpi-odd*:
 fixes k::int
 assumes *odd k*
 shows *sin (pi / 2 + real k * pi) = -1*

$\langle proof \rangle$

lemma *sin-pi2-kpi-even*:

fixes $k::int$

assumes *even k*

shows $\sin (pi / 2 + \text{real } k * pi) = 1$

$\langle proof \rangle$

lemma *cos-zero-iff-int*:

shows $\cos x = 0 \longleftrightarrow (\exists k::int. \text{odd } k \wedge x = \text{real } k * (pi / 2))$

$\langle proof \rangle$

lemma *sin-zero-iff-int*:

$\sin x = 0 \longleftrightarrow (\exists k::int. \text{even } k \wedge x = \text{real } k * (pi / 2))$

$\langle proof \rangle$

lemma *cos0-sin1*:

assumes $\cos \varphi = 0 \sin \varphi = 1$

shows $\exists k::int. \varphi = pi/2 + 2*k*pi$

$\langle proof \rangle$

lemma *cos-0-iff-normalized*:

assumes $\cos \varphi = 0 \ -pi < \varphi \leq pi$

shows $\varphi = pi/2 \vee \varphi = -pi/2$

$\langle proof \rangle$

lemma *sin-0-iff-normalized*:

assumes $\sin \varphi = 0 \ -pi < \varphi \leq pi$

shows $\varphi = 0 \vee \varphi = pi$

$\langle proof \rangle$

lemma *cos1-sin0*:

assumes $\cos \varphi = 1 \sin \varphi = 0$

shows $\exists k::int. \varphi = 2*k*pi$

$\langle proof \rangle$

lemma *sin-cos-eq*:

fixes $a b :: real$

assumes $\cos a = \cos b \sin a = \sin b$

shows $\exists k::int. a - b = 2*k*pi$

$\langle proof \rangle$

lemma *sin-monotone-2pi*: **assumes** $-(pi / 2) \leq y$ **and** $y < x$ **and** $x \leq pi / 2$

shows $\sin y < \sin x$

$\langle proof \rangle$

lemma *sin-inj*:

assumes $\alpha \neq \alpha' - \pi/2 \leq \alpha \wedge \alpha \leq \pi/2 - \pi/2 \leq \alpha' \wedge \alpha' \leq \pi/2$
shows $\sin \alpha \neq \sin \alpha'$
 $\langle \text{proof} \rangle$

lemma *arccos-le-pi2*:
assumes $a \geq 0 \wedge a \leq 1$
shows $\arccos a \leq \pi/2$
 $\langle \text{proof} \rangle$

definition *atan2* **where**
 $\text{atan2 } y \ x =$
 $(\text{if } x > 0 \text{ then } \arctan (y/x)$
 $\text{else if } x < 0 \text{ then}$
 $\quad \text{if } y > 0 \text{ then } \arctan (y/x) + \pi \text{ else } \arctan (y/x) - \pi$
 else
 $\quad \text{if } y > 0 \text{ then } \pi/2 \text{ else if } y < 0 \text{ then } -\pi/2 \text{ else } 0)$

lemma *atan2-bounded*: $-\pi \leq \text{atan2 } y \ x \wedge \text{atan2 } y \ x < \pi$
 $\langle \text{proof} \rangle$

lemma *cos-periodic-nat[simp]*: **fixes** $n :: \text{nat}$ **shows** $\cos (x + n * (2 * \pi)) = \cos x$
 $\langle \text{proof} \rangle$

lemma *cos-periodic-int[simp]*: **fixes** $i :: \text{int}$ **shows** $\cos (x + i * (2 * \pi)) = \cos x$
 $\langle \text{proof} \rangle$

abbreviation *canon-ang-P* **where**
 $\text{canon-ang-P } \alpha \ \alpha' \equiv (-\pi < \alpha' \wedge \alpha' \leq \pi) \wedge (\exists k :: \text{int}. \alpha - \alpha' = 2 * k * \pi)$

definition *canon-ang* $:: \text{real} \Rightarrow \text{real}$ ($|\cdot|$) **where**
 $|\alpha| = (\text{THE } \alpha'. \text{canon-ang-P } \alpha \ \alpha')$

lemma *canon-ang-ex*:
shows $\exists \alpha'. \text{canon-ang-P } \alpha \ \alpha'$
 $\langle \text{proof} \rangle$

lemma *canon-ang-unique*:
assumes $\text{canon-ang-P } \alpha \ \alpha' \wedge \text{canon-ang-P } \alpha \ \alpha''$
shows $\alpha' = \alpha''$
 $\langle \text{proof} \rangle$

lemma *canon-ang*:

$-pi < \lfloor \alpha \rfloor \mid \alpha \rfloor \leq pi \exists k::int. \alpha - \lfloor \alpha \rfloor = 2*k*pi$
 $\langle proof \rangle$

lemma *canon-ang-id*:
assumes $-pi < \alpha \wedge \alpha \leq pi$
shows $\lfloor \alpha \rfloor = \alpha$
 $\langle proof \rangle$

lemma *canon-ang-eq*:
assumes $\exists k::int. \alpha' - \alpha'' = 2*k*pi$
shows $\lfloor \alpha' \rfloor = \lfloor \alpha'' \rfloor$
 $\langle proof \rangle$

lemma *canon-ang-eqI*:
assumes $\exists k::int. \alpha' - \alpha = 2 * k * pi - pi < \alpha' \wedge \alpha' \leq pi$
shows $\lfloor \alpha \rfloor = \alpha'$
 $\langle proof \rangle$

lemma *canon-ang-arg*:
 $\lfloor arg\ z \rfloor = arg\ z$
 $\langle proof \rangle$

lemma *canon-ang-uminus*:
assumes $\lfloor \alpha \rfloor \neq pi$
shows $\lfloor -\alpha \rfloor = -\lfloor \alpha \rfloor$
 $\langle proof \rangle$

lemma *canon-ang-uminus-pi*:
assumes $\lfloor \alpha \rfloor = pi$
shows $\lfloor -\alpha \rfloor = \lfloor \alpha \rfloor$
 $\langle proof \rangle$

lemma *canon-ang-diff*:
 $\lfloor \alpha - \beta \rfloor = \lfloor \lfloor \alpha \rfloor - \lfloor \beta \rfloor \rfloor$
 $\langle proof \rangle$

lemma *canon-ang-sum*:
 $\lfloor \alpha + \beta \rfloor = \lfloor \lfloor \alpha \rfloor + \lfloor \beta \rfloor \rfloor$
 $\langle proof \rangle$

lemma *canon-ang-plus-pi1*:
assumes $0 < \alpha \wedge \alpha \leq 2*pi$
shows $\lfloor \alpha + pi \rfloor = \alpha - pi$
 $\langle proof \rangle$

lemma *canon-ang-plus-pi2*:
assumes $-2*pi < \alpha \wedge \alpha \leq 0$
shows $\lfloor \alpha + pi \rfloor = \alpha + pi$
 $\langle proof \rangle$

lemma *canon-ang-minus-pi1*:

assumes $0 < \alpha \leq 2\pi$

shows $\lfloor \alpha - \pi \rfloor = \alpha - \pi$

<proof>

lemma *canon-ang-minus-pi2*:

assumes $-2\pi < \alpha \leq 0$

shows $\lfloor \alpha - \pi \rfloor = \alpha + \pi$

<proof>

lemma *[simp]*: $\lfloor 0 \rfloor = 0$

<proof>

lemma *canon-ang-cos [simp]*: $\cos \lfloor \alpha \rfloor = \cos \alpha$

<proof>

lemma *[simp]*: $\text{cis } \varphi * \text{cis } (-\varphi) = 1$

<proof>

lemma *cis-eq*:

assumes $\text{cis } a = \text{cis } b$

shows $\exists k::\text{int}. a - b = 2 * k * \pi$

<proof>

lemma *cis-inj*:

assumes $\text{cis } \alpha = \text{cis } \alpha' - \pi < \alpha \leq \pi - \pi < \alpha' \leq \pi$

shows $\alpha = \alpha'$

<proof>

lemma *re-complex-zero-arg1*:

assumes $\arg z = \pi/2 \vee \arg z = -\pi/2$

shows $\text{Re } z = 0$

<proof>

lemma *re-complex-zero-arg2*:

assumes $\text{Re } z = 0 \wedge z \neq 0$

shows $\arg z = \pi/2 \vee \arg z = -\pi/2$

<proof>

lemma *im-complex-zero-arg1*:

assumes $\arg z = 0 \vee \arg z = \pi$

shows $\text{Im } z = 0$

<proof>

lemma *im-complex-zero-arg2*:
 assumes $\text{Im } z = 0$
 shows $\arg z = 0 \vee \arg z = \pi$
 $\langle \text{proof} \rangle$

lemma *arg-complex-of-real-positive*:
 assumes $k > 0$
 shows $\arg (\text{cor } k) = 0$
 $\langle \text{proof} \rangle$

lemma *arg-complex-of-real-negative*:
 assumes $k < 0$
 shows $\arg (\text{cor } k) = \pi$
 $\langle \text{proof} \rangle$

lemma
 $[\text{simp}]$: $\arg ii = \pi/2$
 $\langle \text{proof} \rangle$

lemma
 $[\text{simp}]$: $\arg (-ii) = -\pi/2$
 $\langle \text{proof} \rangle$

lemma *arg-cis*:
 shows $\arg (\text{cis } \varphi) = |\varphi|$
 $\langle \text{proof} \rangle$

lemma *cos-arg*:
 assumes $z \neq 0$
 shows $\cos (\arg z) = \text{Re } z / \text{cmod } z$
 $\langle \text{proof} \rangle$

lemma *sin-arg*:
 assumes $z \neq 0$
 shows $\sin (\arg z) = \text{Im } z / \text{cmod } z$
 $\langle \text{proof} \rangle$

lemma *cis-arg-mult*:
 assumes $a * z \neq 0$
 shows $\text{cis } (\arg (a * z)) = \text{cis } (\arg a + \arg z)$
 $\langle \text{proof} \rangle$

lemma *arg-mult-2kpi*:
 assumes $a * z \neq 0$
 shows $\exists k::\text{int}. \arg (a * z) = \arg a + \arg z + 2*k*\pi$
 $\langle \text{proof} \rangle$

lemma *arg-mult*:

assumes $z1 * z2 \neq 0$
shows $\arg(z1 * z2) = \lfloor \arg z1 + \arg z2 \rfloor$
 $\langle \text{proof} \rangle$

lemma *arg-mult-real-positive*:
assumes $k > 0$
shows $\arg(\text{cor } k * z) = \arg z$
 $\langle \text{proof} \rangle$

lemma *arg-mult-real-negative*:
assumes $k < 0$
shows $\arg(\text{cor } k * z) = \arg(-z)$
 $\langle \text{proof} \rangle$

lemma *arg-cnj1*:
assumes $\arg z = \pi$
shows $\arg(\text{cnj } z) = \pi$
 $\langle \text{proof} \rangle$

lemma *arg-cnj2*:
assumes $\arg z \neq \pi$
shows $\arg(\text{cnj } z) = -\arg z$
 $\langle \text{proof} \rangle$

lemma *arg-div-real-positive*:
assumes $k \neq 0 \ k > 0$
shows $\arg(z / \text{cor } k) = \arg z$
 $\langle \text{proof} \rangle$

lemma *arg-inv1*:
assumes $z \neq 0 \ \arg z \neq \pi$
shows $\arg(1 / z) = -\arg z$
 $\langle \text{proof} \rangle$

lemma *arg-inv2*:
assumes $z \neq 0 \ \arg z = \pi$
shows $\arg(1 / z) = \pi$
 $\langle \text{proof} \rangle$

lemma *arg-inv-2kpi*:
assumes $z \neq 0$
shows $\exists k::\text{int. } \arg(1 / z) = -\arg z + 2*k*\pi$
 $\langle \text{proof} \rangle$

lemma *arg-inv*:
assumes $z \neq 0$

shows $\arg (1 / z) = \lfloor - \arg z \rfloor$
 $\langle \text{proof} \rangle$

lemma *arg-div-2kpi*:
assumes $z1 \neq 0 \ z2 \neq 0$
shows $\exists k::\text{int}. \arg (z1 / z2) = \arg z1 - \arg z2 + 2*k*\pi$
 $\langle \text{proof} \rangle$

lemma *arg-div*:
assumes $z1 \neq 0 \ z2 \neq 0$
shows $\arg(z1 / z2) = \lfloor \arg z1 - \arg z2 \rfloor$
 $\langle \text{proof} \rangle$

lemma *arg-uminus*:
assumes $z \neq 0$
shows $\arg (-z) = \lfloor \arg z + \pi \rfloor$
 $\langle \text{proof} \rangle$

definition
 $\text{csqrt } z = \text{rcis } (\text{sqrt } (\text{cmod } z)) (\arg z / 2)$

lemma [*simp*]: $(\text{csqrt } x)^2 = x$
 $\langle \text{proof} \rangle$

lemma *ex-complex-sqrt*: $\exists s::\text{complex}. s*s = z$
 $\langle \text{proof} \rangle$

lemma *csqrt*:
assumes $s * s = z$
shows $s = \text{csqrt } z \vee s = -\text{csqrt } z$
 $\langle \text{proof} \rangle$

lemma [*simp*]: $\text{csqrt } x = 0 \longleftrightarrow x = 0$
 $\langle \text{proof} \rangle$

lemma *csqrt-mult*: $\text{csqrt } (a * b) = \text{csqrt } a * \text{csqrt } b \vee \text{csqrt } (a * b) = - \text{csqrt } a * \text{csqrt } b$
 $\langle \text{proof} \rangle$

lemma *csqrt-real*:
assumes *is-real* x
shows $(\text{Re } x \geq 0 \wedge \text{csqrt } x = \text{cor } (\text{sqrt } (\text{Re } x))) \vee$

$(Re\ x < 0 \wedge csqrt\ x = ii * cor\ (sqrt\ (-\ (Re\ x))))$
 $\langle proof \rangle$

lemma *is-real-rot-to-xaxis*:
assumes $z \neq 0$
shows *is-real* $(cis\ (-arg\ z) * z)$
 $\langle proof \rangle$

lemma *cmod-1-plus-mult-le*:
 $cmod\ (1 + z * w) \leq sqrt((1 + (cmod\ z)^2) * (1 + (cmod\ w)^2))$
 $\langle proof \rangle$

lemma *cmod-diff-ge*: $cmod\ (b - c) \geq sqrt\ (1 + (cmod\ b)^2) - sqrt\ (1 + (cmod\ c)^2)$
 $\langle proof \rangle$

lemma *cmod-diff-le*: $cmod\ (b - c) \leq sqrt\ (1 + (cmod\ b)^2) + sqrt\ (1 + (cmod\ c)^2)$
 $\langle proof \rangle$

definition *cdist where*
 $[simp]:\ cdist\ z1\ z2 \equiv cmod\ (z2 - z1)$

lemma $[simp]$:
fixes $z1\ z2 :: complex$
assumes $z1 \neq 0\ z2 \neq 0$
shows $\exists k. k \neq 0 \wedge z2 = k * z1$
 $\langle proof \rangle$

lemma $[simp]$:
fixes $z :: complex$
assumes $z \neq 0$
shows $\exists k. k \neq 0 \wedge k * z = 1$
 $\langle proof \rangle$

lemma $[simp]$:
fixes $z :: complex$
shows $\exists k. k \neq 0 \wedge k * z = z$
 $\langle proof \rangle$

end

2 Systems of linear equations

theory *LinearSystems*
imports *MoreComplex*
begin

definition *det2* **where**

[*simp*]: $\text{det2 } a11 \ a12 \ a21 \ a22 \equiv a11 * a22 - a12 * a21$

lemma *regular-homogenous-system*:

fixes *a11::complex*

assumes $a11 * a22 - a12 * a21 \neq 0$ $a11 * x1 + a12 * x2 = 0$ $a21 * x1 + a22 * x2 = 0$

shows $x1 = 0 \wedge x2 = 0$

<proof>

lemma *regular-system*:

fixes *a11::complex*

assumes $a11 * a22 - a12 * a21 \neq 0$

shows $\exists! x.$

$a11 * (\text{fst } x) + a12 * (\text{snd } x) = b1 \wedge$

$a21 * (\text{fst } x) + a22 * (\text{snd } x) = b2$

<proof>

lemma *singular-system*:

fixes *a11::complex*

assumes $a11 * a22 - a12 * a21 = 0$ $a11 \neq 0 \vee a12 \neq 0$

assumes *: $a11 * \text{fst } x0 + a12 * \text{snd } x0 = b1$ $a21 * \text{fst } x0 + a22 * \text{snd } x0 = b2$

assumes **: $a11 * \text{fst } x + a12 * \text{snd } x = b1$

shows $a21 * \text{fst } x + a22 * \text{snd } x = b2$

<proof>

lemma *cnj-equation*:

assumes $a * z1 + b * z2 = c$

shows $\text{cnj } a * \text{cnj } z1 + \text{cnj } b * \text{cnj } z2 = \text{cnj } c$

<proof>

lemma *regular-cnj-system*:

assumes $\text{det2 } a1 \ (\text{cnj } a1) \ a2 \ (\text{cnj } a2) \neq 0$ *is-real b1 is-real b2*

shows $\exists! \mu. a1 * \text{cnj } \mu + \text{cnj } a1 * \mu = b1 \wedge$

$a2 * \text{cnj } \mu + \text{cnj } a2 * \mu = b2$

<proof>

end

3 Quadratic equations

```
theory Quadratic
imports Complex MoreComplex
begin
```

```
lemma real-quadratic-equation:
  fixes  $\xi :: \text{real}$ 
  assumes  $\xi^2 + b * \xi + c = 0$   $b^2 - 4*c \geq 0$ 
  shows  $\xi = (-b + \text{sqrt}(b^2 - 4*c)) / 2 \vee \xi = (-b - \text{sqrt}(b^2 - 4*c)) / 2$ 
  <proof>
```

```
lemma real-quadratic-equation':
  fixes  $\xi :: \text{real}$ 
  assumes  $b^2 - 4*c \geq 0$   $\xi = (-b + \text{sqrt}(b^2 - 4*c)) / 2 \vee \xi = (-b - \text{sqrt}(b^2 - 4*c)) / 2$ 
  shows  $\xi^2 + b * \xi + c = 0$ 
  <proof>
```

```
lemma complex-quadratic-equation:
  fixes  $\xi :: \text{complex}$ 
  assumes  $\xi^2 + b * \xi + c = 0$ 
  shows  $\xi = (-b + \text{csqrt}(b^2 - 4*c)) / 2 \vee \xi = (-b - \text{csqrt}(b^2 - 4*c)) / 2$ 
  <proof>
```

```
lemma complex-quadratic-equation':
  fixes  $\xi :: \text{complex}$ 
  assumes  $\xi = (-b + \text{csqrt}(b^2 - 4*c)) / 2 \vee$ 
          $\xi = (-b - \text{csqrt}(b^2 - 4*c)) / 2$ 
  shows  $\xi^2 + b * \xi + c = 0$ 
  <proof>
```

```
lemma complex-quadratic-equation-full:
  fixes  $\xi :: \text{complex}$ 
  assumes  $a*\xi^2 + b * \xi + c = 0$   $a \neq 0$ 
  shows  $\xi = (-b + \text{csqrt}(b^2 - 4*a*c)) / (2*a) \vee$ 
          $\xi = (-b - \text{csqrt}(b^2 - 4*a*c)) / (2*a)$ 
  <proof>
```

```
lemma complex-quadratic-two-solutions:
  fixes  $b\ c :: \text{complex}$ 
  assumes  $b^2 - 4*c \neq 0$ 
  shows  $\exists\ k_1\ k_2. k_1 \neq k_2 \wedge k_1^2 + b*k_1 + c = 0 \wedge k_2^2 + b*k_2 + c = 0$ 
  <proof>
```

```
end
```

4 Vectors, Matrices

```
theory Matrices
imports MoreComplex LinearSystems Quadratic
begin
```

4.1 Vectors

Type of complex vector

```
type-synonym complex-vec = complex × complex
```

```
definition vec-zero :: complex-vec where
  [simp]: vec-zero = (0, 0)
```

Vector scalar multiplication

```
fun mult-sv :: complex ⇒ complex-vec ⇒ complex-vec (infixl *sv 100) where
  k *sv (x, y) = (k*x, k*y)
```

```
lemma fst-mult-sv [simp]: fst (k *sv v) = k * fst v
⟨proof⟩
```

```
lemma snd-mult-sv [simp]: snd (k *sv v) = k * snd v
⟨proof⟩
```

```
lemma mult-sv-mult-sv [simp]: k1 *sv (k2 *sv v) = (k1*k2) *sv v
⟨proof⟩
```

```
lemma one-mult-sv [simp]: 1 *sv v = v
⟨proof⟩
```

Multiplication of two vectors

```
fun mult-vv :: complex × complex ⇒ complex × complex ⇒ complex (infixl *vv
100) where
  (x, y) *vv (a, b) = x*a + y*b
```

```
lemma mult-vv-commute: v1 *vv v2 = v2 *vv v1
⟨proof⟩
```

```
lemma mult-vv-scale-sv1:
  (k *sv v1) *vv v2 = k * (v1 *vv v2)
⟨proof⟩
```

```
lemma mult-vv-scale-sv2:
  v1 *vv (k *sv v2) = k * (v1 *vv v2)
⟨proof⟩
```

Conjugate vector

```
fun vec-map where
```

$vec-map\ f\ (x, y) = (f\ x, f\ y)$

definition $vec-cnj$ **where** $vec-cnj = vec-map\ conj$

lemma $vec-cnj-vec-cnj\ [simp]:\ vec-cnj\ (vec-cnj\ v) = v$
 $\langle proof \rangle$

lemma $cnj-mult-vv: conj\ (v1\ *_v\ v2) = (vec-cnj\ v1)\ *_v\ (vec-cnj\ v2)$
 $\langle proof \rangle$

lemma $vec-cnj-sv\ [simp]:\ vec-cnj\ (k\ *_s\ A) = conj\ k\ *_s\ vec-cnj\ A$
 $\langle proof \rangle$

lemma $scalsquare-vv-zero:$
 $(vec-cnj\ v)\ *_v\ v = 0 \longleftrightarrow v = vec-zero$
 $\langle proof \rangle$

4.2 Matrices

Type of complex matrices

type-synonym $complex-mat = complex \times complex \times complex \times complex$

Matrix scalar multiplication

fun $mult-sm :: complex \Rightarrow complex-mat \Rightarrow complex-mat$ (**infixl** $*_{sm}$ 100) **where**
 $k\ *_{sm}\ (a, b, c, d) = (k*a, k*b, k*c, k*d)$

lemma $[simp]:\ k1\ *_{sm}\ (k2\ *_{sm}\ A) = (k1*k2)\ *_{sm}\ A$
 $\langle proof \rangle$

lemma $[simp]:\ 1\ *_{sm}\ A = A$
 $\langle proof \rangle$

lemma $mult-sm-inv-l:$
assumes $k \neq 0$ $k\ *_{sm}\ A = B$
shows $A = (1/k)\ *_{sm}\ B$
 $\langle proof \rangle$

Matrix addition and subtraction

definition $mat-zero :: complex-mat$ **where** $[simp]:\ mat-zero = (0, 0, 0, 0)$

fun $mat-plus :: complex-mat \Rightarrow complex-mat \Rightarrow complex-mat$ (**infixl** $+_{mm}$ 100)
where
 $mat-plus\ (a1, b1, c1, d1)\ (a2, b2, c2, d2) = (a1+a2, b1+b2, c1+c2, d1+d2)$

fun $mat-minus :: complex-mat \Rightarrow complex-mat \Rightarrow complex-mat$ (**infixl** $-_{mm}$ 100)
where
 $mat-minus\ (a1, b1, c1, d1)\ (a2, b2, c2, d2) = (a1-a2, b1-b2, c1-c2, d1-d2)$

fun *mat-uminus* :: *complex-mat* \Rightarrow *complex-mat* **where**
mat-uminus (*a*, *b*, *c*, *d*) = ($-a$, $-b$, $-c$, $-d$)

lemma *nonzero-mult-real*:
assumes $A \neq \text{mat-zero}$ $k \neq 0$
shows $k *_{sm} A \neq \text{mat-zero}$
 $\langle \text{proof} \rangle$

Matrix multiplication

fun *mult-mm* :: *complex-mat* \Rightarrow *complex-mat* \Rightarrow *complex-mat* (**infixl** $*_{mm}$ 100)
where
 $(a1, b1, c1, d1) *_{mm} (a2, b2, c2, d2) =$
 $(a1*a2 + b1*c2, a1*b2 + b1*d2, c1*a2 + d1*c2, c1*b2 + d1*d2)$

lemma *mult-mm-assoc*: $A *_{mm} (B *_{mm} C) = (A *_{mm} B) *_{mm} C$
 $\langle \text{proof} \rangle$

lemma *mult-assoc-5*: $A *_{mm} (B *_{mm} C *_{mm} D) *_{mm} E = (A *_{mm} B) *_{mm} C$
 $*_{mm} (D *_{mm} E)$
 $\langle \text{proof} \rangle$

lemma *mat-zero-r* [*simp*]: $A *_{mm} \text{mat-zero} = \text{mat-zero}$
 $\langle \text{proof} \rangle$

lemma *mat-zero-l* [*simp*]: $\text{mat-zero} *_{mm} A = \text{mat-zero}$
 $\langle \text{proof} \rangle$

definition *eye* :: *complex-mat* **where**
 $[\text{simp}]: \text{eye} = (1, 0, 0, 1)$

lemma *mat-eye-l*:
 $\text{eye} *_{mm} A = A$
 $\langle \text{proof} \rangle$

lemma *mat-eye-r*:
 $A *_{mm} \text{eye} = A$
 $\langle \text{proof} \rangle$

lemma *mult-mm-sm* [*simp*]: $A *_{mm} (k *_{sm} B) = k *_{sm} (A *_{mm} B)$
 $\langle \text{proof} \rangle$

lemma *mult-sm-mm* [*simp*]: $(k *_{sm} A) *_{mm} B = k *_{sm} (A *_{mm} B)$
 $\langle \text{proof} \rangle$

lemma *mult-sm-eye-mm* [*simp*]: $k *_{sm} \text{eye} *_{mm} A = k *_{sm} A$

$\langle proof \rangle$

Matrix determinant

fun *mat-det* **where** *mat-det* (*a*, *b*, *c*, *d*) = *a*d* - *b*c*

lemma *mat-det-mult* [*simp*]: *mat-det* (*A* *_{mm} *B*) = *mat-det* *A* * *mat-det* *B*
 $\langle proof \rangle$

lemma *mat-det-mult-sm* [*simp*]: *mat-det* (*k* *_{sm} *A*) = (*k*k*) * *mat-det* *A*
 $\langle proof \rangle$

Matrix inverse

fun *mat-inv* :: *complex-mat* \Rightarrow *complex-mat* **where**
 mat-inv (*a*, *b*, *c*, *d*) = (1/(*a*d* - *b*c*)) *_{sm} (*d*, -*b*, -*c*, *a*)

lemma *mat-inv-r*:
 assumes *mat-det* *A* \neq 0
 shows *A* *_{mm} (*mat-inv* *A*) = *eye*
 $\langle proof \rangle$

lemma *mat-inv-l*:
 assumes *mat-det* *A* \neq 0
 shows (*mat-inv* *A*) *_{mm} *A* = *eye*
 $\langle proof \rangle$

lemma *mat-det-inv*:
 assumes *mat-det* *A* \neq 0
 shows *mat-det* (*mat-inv* *A*) = 1 / *mat-det* *A*
 $\langle proof \rangle$

lemma *mult-mm-inv-l*:
 assumes *mat-det* *A* \neq 0 *A* *_{mm} *B* = *C*
 shows *B* = *mat-inv* *A* *_{mm} *C*
 $\langle proof \rangle$

lemma *mult-mm-inv-r*:
 assumes *mat-det* *B* \neq 0 *A* *_{mm} *B* = *C*
 shows *A* = *C* *_{mm} *mat-inv* *B*
 $\langle proof \rangle$

lemma *mult-mm-non-zero-l*:
 assumes *mat-det* *A* \neq 0 *B* \neq *mat-zero*
 shows *A* *_{mm} *B* \neq *mat-zero*
 $\langle proof \rangle$

lemma *mat-inv-mult-mm*:
 assumes *mat-det* *A* \neq 0 *mat-det* *B* \neq 0
 shows *mat-inv* (*A* *_{mm} *B*) = *mat-inv* *B* *_{mm} *mat-inv* *A*
 $\langle proof \rangle$

lemma *mult-mm-cancel-l*:
 assumes *mat-det* $M \neq 0$ $M *_{mm} A = M *_{mm} B$
 shows $A = B$
 $\langle proof \rangle$

lemma *mult-mm-cancel-r*:
 assumes *mat-det* $M \neq 0$ $A *_{mm} M = B *_{mm} M$
 shows $A = B$
 $\langle proof \rangle$

lemma *mult-mm-non-zero-r*:
 assumes $A \neq \text{mat-zero}$ *mat-det* $B \neq 0$
 shows $A *_{mm} B \neq \text{mat-zero}$
 $\langle proof \rangle$

lemma *mat-inv-mult-sm*:
 assumes $k \neq 0$
 shows *mat-inv* $(k *_{sm} A) = (1 / k) *_{sm} \text{mat-inv } A$
 $\langle proof \rangle$

lemma *mat-inv-inv [simp]*:
 assumes *mat-det* $M \neq 0$
 shows *mat-inv* (*mat-inv* M) = M
 $\langle proof \rangle$

Matrix transpose

fun *mat-transpose* **where** *mat-transpose* $(a, b, c, d) = (a, c, b, d)$

lemma *[simp]*: *mat-transpose* (*mat-transpose* A) = A
 $\langle proof \rangle$

lemma *[simp]*: *mat-transpose* $(k *_{sm} A) = k *_{sm} (\text{mat-transpose } A)$
 $\langle proof \rangle$

lemma *[simp]*: *mat-transpose* $(A *_{mm} B) = \text{mat-transpose } B *_{mm} \text{mat-transpose } A$
 $\langle proof \rangle$

lemma *mat-inv-transpose*: *mat-transpose* (*mat-inv* M) = *mat-inv* (*mat-transpose* M)
 $\langle proof \rangle$

lemma *mat-det-transpose*:
 fixes $M :: \text{complex-mat}$
 shows *[simp]*: *mat-det* (*mat-transpose* M) = *mat-det* M
 $\langle proof \rangle$

Diagonal matrices

fun *mat-diagonal* **where**
mat-diagonal (*A*, *B*, *C*, *D*) = (*B* = 0 ∧ *C* = 0)

Matrix conjugate

fun *mat-map* **where**
mat-map *f* (*a*, *b*, *c*, *d*) = (*f a*, *f b*, *f c*, *f d*)

definition *mat-cnj* **where** *mat-cnj* = *mat-map* *cnj*

lemma [*simp*]: *mat-cnj* (*mat-cnj A*) = *A*
 ⟨*proof*⟩

lemma *mat-cnj-sm* [*simp*]: *mat-cnj* (*k* *_{sm} *A*) = *cnj k* *_{sm} (*mat-cnj A*)
 ⟨*proof*⟩

lemma *mat-det-cnj* [*simp*]: *mat-det* (*mat-cnj A*) = *cnj* (*mat-det A*)
 ⟨*proof*⟩

lemma *nonzero-mat-cnj*: *mat-cnj A* = *mat-zero* ↔ *A* = *mat-zero*
 ⟨*proof*⟩

lemma *mat-inv-cnj*: *mat-cnj* (*mat-inv M*) = *mat-inv* (*mat-cnj M*)
 ⟨*proof*⟩

Matrix adjoint (conjugate

definition *mat-adj* **where** *mat-adj A* = *mat-cnj* (*mat-transpose A*)

lemma *mat-adj-mult-mm* [*simp*]: *mat-adj* (*A* *_{mm} *B*) = *mat-adj B* *_{mm} *mat-adj A*
 ⟨*proof*⟩

lemma *mat-adj-mult-sm* [*simp*]: *mat-adj* (*k* *_{sm} *A*) = *cnj k* *_{sm} *mat-adj A*
 ⟨*proof*⟩

lemma *mat-det-adj*: *mat-det* (*mat-adj A*) = *cnj* (*mat-det A*)
 ⟨*proof*⟩

lemma *mat-adj-inv*:
assumes *mat-det M* ≠ 0
shows *mat-adj* (*mat-inv M*) = *mat-inv* (*mat-adj M*)
 ⟨*proof*⟩

lemma *mat-transpose-mat-cnj*: *mat-transpose* (*mat-cnj A*) = *mat-adj A*
 ⟨*proof*⟩

lemma [*simp*]: *mat-adj* (*mat-adj A*) = *A*
 ⟨*proof*⟩

Matrix trace

fun *mat-trace* **where**

mat-trace (*a*, *b*, *c*, *d*) = *a* + *d*

Multiplication of matrix and a vector

fun *mult-mv* :: *complex-mat* \Rightarrow *complex-vec* \Rightarrow *complex-vec* (**infixl** \ast_{mv} 100) **where**

(*a*, *b*, *c*, *d*) \ast_{mv} (*x*, *y*) = (*x***a* + *y***b*, *x***c* + *y***d*)

fun *mult-vm* :: *complex-vec* \Rightarrow *complex-mat* \Rightarrow *complex-vec* (**infixl** \ast_{vm} 100) **where**

(*x*, *y*) \ast_{vm} (*a*, *b*, *c*, *d*) = (*x***a* + *y***c*, *x***b* + *y***d*)

lemma *eye-mv-l* [*simp*]: *eye* \ast_{mv} *v* = *v*

\langle *proof* \rangle

lemma *mult-mv-mv* [*simp*]: *B* \ast_{mv} (*A* \ast_{mv} *v*) = (*B* \ast_{mm} *A*) \ast_{mv} *v*

\langle *proof* \rangle

lemma *mult-vm-vm* [*simp*]: (*v* \ast_{vm} *A*) \ast_{vm} *B* = *v* \ast_{vm} (*A* \ast_{mm} *B*)

\langle *proof* \rangle

lemma *mult-mv-inv*:

assumes *x* = *A* \ast_{mv} *y* *mat-det* *A* \neq 0

shows *y* = (*mat-inv* *A*) \ast_{mv} *x*

\langle *proof* \rangle

lemma *mult-vm-inv*:

assumes *x* = *y* \ast_{vm} *A* *mat-det* *A* \neq 0

shows *y* = *x* \ast_{vm} (*mat-inv* *A*)

\langle *proof* \rangle

lemma *mult-mv-cancel-l*:

assumes *mat-det* *A* \neq 0 *A* \ast_{mv} *v* = *A* \ast_{mv} *v'*

shows *v* = *v'*

\langle *proof* \rangle

lemma *mult-vm-cancel-r*:

assumes *mat-det* *A* \neq 0 *v* \ast_{vm} *A* = *v'* \ast_{vm} *A*

shows *v* = *v'*

\langle *proof* \rangle

lemma *vec-zero-l* [*simp*]:

A \ast_{mv} *vec-zero* = *vec-zero*

\langle *proof* \rangle

lemma *vec-zero-r* [*simp*]:

vec-zero \ast_{vm} *A* = *vec-zero*

\langle *proof* \rangle

lemma *mult-mv-nonzero*:

assumes *v* \neq *vec-zero* *mat-det* *A* \neq 0

shows $A *_{mv} v \neq \text{vec-zero}$
 $\langle \text{proof} \rangle$

lemma *mult-vm-nonzero*:
assumes $v \neq \text{vec-zero}$ $\text{mat-det } A \neq 0$
shows $v *_{vm} A \neq \text{vec-zero}$
 $\langle \text{proof} \rangle$

lemma *mult-sv-mv*: $k *_{sv} (A *_{mv} v) = (A *_{mv} (k *_{sv} v))$
 $\langle \text{proof} \rangle$

lemma *mult-mv-mult-vm*: $A *_{mv} x = x *_{vm} (\text{mat-transpose } A)$
 $\langle \text{proof} \rangle$

lemma
mult-mv-vv: $A *_{mv} v1 *_{vv} v2 = v1 *_{vv} (\text{mat-transpose } A *_{mv} v2)$
 $\langle \text{proof} \rangle$

lemma *mult-vv-mv*: $x *_{vv} (A *_{mv} y) = (x *_{vm} A) *_{vv} y$
 $\langle \text{proof} \rangle$

lemma *vec-cnj-mult-mv*:
shows $\text{vec-cnj } (A *_{mv} x) = (\text{mat-cnj } A) *_{mv} (\text{vec-cnj } x)$
 $\langle \text{proof} \rangle$

lemma *vec-cnj-mult-vm*: $\text{vec-cnj } (v *_{vm} A) = \text{vec-cnj } v *_{vm} \text{mat-cnj } A$
 $\langle \text{proof} \rangle$

4.3 Eigenvalues and eigenvectors

definition *eigenpair* **where**
 $[\text{simp}]: \text{eigenpair } k \ v \ H \longleftrightarrow v \neq \text{vec-zero} \wedge H *_{mv} v = k *_{sv} v$

definition *eigenval* **where**
 $[\text{simp}]: \text{eigenval } k \ H \longleftrightarrow (\exists \ v. v \neq \text{vec-zero} \wedge H *_{mv} v = k *_{sv} v)$

lemma *eigen-equation*:
shows $\text{eigenval } k \ H \longleftrightarrow k^2 - \text{mat-trace } H * k + \text{mat-det } H = 0$ (**is ?lhs** \longleftrightarrow **?rhs**)
 $\langle \text{proof} \rangle$

4.4 Bilinear and Quadratic forms; Congruence

Bilinear forms

definition *bilinear-form* **where**
 $[\text{simp}]: \text{bilinear-form } v1 \ v2 \ H = (\text{vec-cnj } v1) *_{vm} H *_{vv} v2$

lemma *bilinear-form-scale-m*:
shows $\text{bilinear-form } v1 \ v2 \ (k *_{sm} H) = k * \text{bilinear-form } v1 \ v2 \ H$

$\langle \text{proof} \rangle$

lemma *bilinear-form-scale-v1*:

shows *bilinear-form* $(k *_{sv} v1) v2 H = \text{cnj } k * \text{bilinear-form } v1 v2 H$
 $\langle \text{proof} \rangle$

lemma *bilinear-form-scale-v2*:

shows *bilinear-form* $v1 (k *_{sv} v2) H = k * \text{bilinear-form } v1 v2 H$
 $\langle \text{proof} \rangle$

Quadratic forms

definition *quad-form where*

$[simp]: \text{quad-form } v H = (\text{vec-cnj } v) *_{vm} H *_{vv} v$

lemma *quad-form* $v H = \text{bilinear-form } v v H$

$\langle \text{proof} \rangle$

lemma *quad-form-scale-v*:

shows *quad-form* $(k *_{sv} v) H = \text{cor } ((\text{cmod } k)^2) * \text{quad-form } v H$
 $\langle \text{proof} \rangle$

lemma *quad-form-scale-m*:

shows *quad-form* $v (k *_{sm} H) = k * \text{quad-form } v H$
 $\langle \text{proof} \rangle$

lemma *cnj-quad-form* $[simp]: \text{cnj } (\text{quad-form } z H) = \text{quad-form } z (\text{mat-adj } H)$

$\langle \text{proof} \rangle$

Matrix congruence

abbreviation *congruence where*

congruence $M H \equiv \text{mat-adj } M *_{mm} H *_{mm} M$

lemma *bilinear-form-congruence*:

assumes *mat-det* $M \neq 0$

shows *bilinear-form* $v1 v2 H = \text{bilinear-form } (M *_{mv} v1) (M *_{mv} v2) (\text{congruence } (\text{mat-inv } M) H)$

$\langle \text{proof} \rangle$

lemma *quad-form-congruence*:

assumes *mat-det* $M \neq 0$

shows *quad-form* $(M *_{mv} z) (\text{congruence } (\text{mat-inv } M) H) = \text{quad-form } z H$
 $\langle \text{proof} \rangle$

lemma *congruence-nonzero*:

assumes $H \neq \text{mat-zero}$ *mat-det* $M \neq 0$

shows *congruence* $M H \neq \text{mat-zero}$
 $\langle \text{proof} \rangle$

lemma *congruence-congruence*:

shows *congruence* $M1$ (*congruence* $M2$ A) = *congruence* ($M2 *_{mm} M1$) A
 ⟨*proof*⟩

lemma [*simp*]: *congruence* *eye* $A = A$
 ⟨*proof*⟩

lemma *congruence-congruence-inv*:
assumes *mat-det* $M \neq 0$
shows *congruence* M (*congruence* (*mat-inv* M) A) = A
 ⟨*proof*⟩

lemma *congruence-inv*:
assumes *mat-det* $M \neq 0$ *congruence* M $A = B$
shows *congruence* (*mat-inv* M) $B = A$
 ⟨*proof*⟩

lemma *congruence-scale-m*:
shows *congruence* A ($k *_{sm} B$) = $k *_{sm}$ (*congruence* A B)
 ⟨*proof*⟩

lemma *inj-congruence*:
assumes *mat-det* $M \neq 0$ *congruence* M $H = \text{congruence } M H'$
shows $H = H'$
 ⟨*proof*⟩

definition *similarity* **where** *similarity* I $M = \text{mat-inv } I *_{mm} M *_{mm} I$

lemma
mat-det-similarity:
assumes *mat-det* $I \neq 0$
shows *mat-det* (*similarity* I M) = *mat-det* M
 ⟨*proof*⟩

lemma *mat-trace-similarity*:
assumes *mat-det* $I \neq 0$
shows *mat-trace* (*similarity* I M) = *mat-trace* M
 ⟨*proof*⟩

end

5 Unitary matrices

theory *UnitaryMatrices*
imports *Matrices*
begin

definition *unitary* **where**

$unitary\ M \longleftrightarrow mat\text{-}adj\ M *_{mm}\ M = eye$

definition *unitary-gen* **where**

$unitary\text{-}gen\ M \longleftrightarrow (\exists\ k::complex. k \neq 0 \wedge mat\text{-}adj\ M *_{mm}\ M = k *_{sm}\ eye)$

lemma *unitary-gen-scale* [simp]:

assumes *unitary-gen* $M\ k \neq 0$

shows *unitary-gen* $(k *_{sm}\ M)$

$\langle proof \rangle$

lemma *unitary-unitary-gen* [simp]: $unitary\ M \implies unitary\text{-}gen\ M$

$\langle proof \rangle$

lemma *unitary-gen-real*:

assumes *unitary-gen* M

shows $(\exists\ k::real. k > 0 \wedge mat\text{-}adj\ M *_{mm}\ M = cor\ k *_{sm}\ eye)$

$\langle proof \rangle$

lemma *unitary-gen-regular*:

assumes *unitary-gen* M

shows $mat\text{-}det\ M \neq 0$

$\langle proof \rangle$

lemmas *unitary-regular* = *unitary-gen-regular*[OF *unitary-unitary-gen*]

lemma

$unitary\text{-}gen\ M \longleftrightarrow (\exists\ k::complex. k \neq 0 \wedge mat\text{-}adj\ M *_{mm}\ (1, 0, 0, 1) *_{mm}\ M = k *_{sm}\ (1, 0, 0, 1))$

$\langle proof \rangle$

lemma *unitary-comp*:

assumes *unitary* $M1\ unitary\ M2$

shows *unitary* $(M1 *_{mm}\ M2)$

$\langle proof \rangle$

lemma *unitary-gen-comp*:

assumes *unitary-gen* $M1\ unitary\text{-}gen\ M2$

shows *unitary-gen* $(M1 *_{mm}\ M2)$

$\langle proof \rangle$

lemma *unitary-adj-eq-inv*:

$unitary\ M \longleftrightarrow mat\text{-}det\ M \neq 0 \wedge mat\text{-}adj\ M = mat\text{-}inv\ M$

$\langle proof \rangle$

lemma *unitary-inv*:

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

lemma *unitary-gen-unitary*:
shows *unitary-gen* $M \longleftrightarrow (\exists k M'. k > 0 \wedge \text{unitary } M' \wedge M = (\text{cor } k *_{sm} \text{eye}) *_{mm} M')$ (**is** $?lhs = ?rhs$)
 $\langle \text{proof} \rangle$

lemma *unitary-gen-inv*:
assumes *unitary-gen* M
shows *unitary-gen* ($\text{mat-inv } M$)
 $\langle \text{proof} \rangle$

lemma *unitary-special*:
assumes *unitary* M $\text{mat-det } M = 1$
shows $\exists a b. M = (a, b, -\text{cnj } b, \text{cnj } a)$
 $\langle \text{proof} \rangle$

lemma *unitary-gen-special*:
assumes *unitary-gen* M $\text{mat-det } M = 1$
shows $\exists a b. M = (a, b, -\text{cnj } b, \text{cnj } a)$
 $\langle \text{proof} \rangle$

lemma *unitary-gen-iff*:
shows *unitary-gen* $M \longleftrightarrow (\exists a b k. k \neq 0 \wedge \text{mat-det } (a, b, -\text{cnj } b, \text{cnj } a) \neq 0 \wedge (M = k *_{sm} (a, b, -\text{cnj } b, \text{cnj } a)))$ (**is** $?lhs = ?rhs$)
 $\langle \text{proof} \rangle$

lemma *unitary-iff*:
shows *unitary* $M \longleftrightarrow$
 $(\exists a b k. (cmod a)^2 + (cmod b)^2 \neq 0 \wedge (cmod k)^2 = 1 / ((cmod a)^2 + (cmod b)^2) \wedge M = k *_{sm} (a, b, -\text{cnj } b, \text{cnj } a))$ (**is** $?lhs = ?rhs$)
 $\langle \text{proof} \rangle$

definition *unitary11 where*
 $\text{unitary11 } M \longleftrightarrow \text{mat-adj } M *_{mm} (1, 0, 0, -1) *_{mm} M = (1, 0, 0, -1)$

definition *unitary11-gen where*
 $\text{unitary11-gen } M \longleftrightarrow (\exists k. k \neq 0 \wedge \text{mat-adj } M *_{mm} (1, 0, 0, -1) *_{mm} M = k *_{sm} (1, 0, 0, -1))$

lemma *unitary11-gen-real*:
 $\text{unitary11-gen } M \longleftrightarrow (\exists k. k \neq 0 \wedge \text{mat-adj } M *_{mm} (1, 0, 0, -1) *_{mm} M = \text{cor } k *_{sm} (1, 0, 0, -1))$

$\langle proof \rangle$

lemma *unitary11-unitary11-gen* [simp]: *unitary11* $M \implies$ *unitary11-gen* M
 $\langle proof \rangle$

lemma *unitary11-gen-regular*:
 assumes *unitary11-gen* M
 shows *mat-det* $M \neq 0$
 $\langle proof \rangle$

lemmas *unitary11-regular* = *unitary11-gen-regular*[OF *unitary11-unitary11-gen*]

lemma *unitary11-gen-mult-sm*:
 assumes $k \neq 0$ *unitary11-gen* M
 shows *unitary11-gen* $(k *_{sm} M)$
 $\langle proof \rangle$

lemma *unitary11-gen-div-sm*:
 assumes $k \neq 0$ *unitary11-gen* $(k *_{sm} M)$
 shows *unitary11-gen* M
 $\langle proof \rangle$

lemma *unitary11-special*:
 assumes *unitary11* M *mat-det* $M = 1$
 shows $\exists a b. M = (a, b, cnj\ b, cnj\ a)$
 $\langle proof \rangle$

lemma *unitary11-gen-special*:
 assumes *unitary11-gen* M *mat-det* $M = 1$
 shows $\exists a b. M = (a, b, cnj\ b, cnj\ a) \vee M = (a, b, -cnj\ b, -cnj\ a)$
 $\langle proof \rangle$

lemma *unitary11-gen-iff'*:
 shows *unitary11-gen* $M \longleftrightarrow$
 $(\exists a\ b\ k. k \neq 0 \wedge \text{mat-det } (a, b, cnj\ b, cnj\ a) \neq 0 \wedge$
 $(M = k *_{sm} (a, b, cnj\ b, cnj\ a) \vee M = k *_{sm} (-1, 0, 0, 1) *_{mm} (a,$
 $b, cnj\ b, cnj\ a)))$ (is ?lhs = ?rhs)
 $\langle proof \rangle$

lemma *unitary11-gen-cis-blaschke*:
 assumes $k \neq 0$ $M = k *_{sm} (a, b, cnj\ b, cnj\ a)$ $a \neq 0$ *mat-det* $(a, b, cnj\ b, cnj\ a) \neq 0$
 shows $\exists k' \varphi a'. k' \neq 0 \wedge a' * cnj\ a' \neq 1 \wedge M = k' *_{sm} (cis\ \varphi, 0, 0, 1) *_{mm}$
 $(1, -a', -cnj\ a', 1)$
 $\langle proof \rangle$

lemma *unitary11-gen-cis-blaschke'*:

assumes $k \neq 0$ $M = k *_{sm} (-1, 0, 0, 1) *_{mm} (a, b, cnj\ b, cnj\ a)$ $a \neq 0$ *mat-det*
 $(a, b, cnj\ b, cnj\ a) \neq 0$
shows $\exists k' \varphi\ a'. k' \neq 0 \wedge a' * cnj\ a' \neq 1 \wedge M = k' *_{sm} (cis\ \varphi, 0, 0, 1) *_{mm}$
 $(1, -a', -cnj\ a', 1)$
 $\langle proof \rangle$

lemma *unitary11-gen-cis-blaschke-rev*:

assumes $k' \neq 0$ $M = k' *_{sm} (cis\ \varphi, 0, 0, 1) *_{mm} (1, -a', -cnj\ a', 1)$ $a' * cnj\ a' \neq 1$
shows $\exists k\ a\ b. k \neq 0 \wedge mat-det\ (a, b, cnj\ b, cnj\ a) \neq 0 \wedge M = k *_{sm} (a, b,$
 $cnj\ b, cnj\ a)$
 $\langle proof \rangle$

lemma *unitary11-gen-cis-inversion*:

assumes $k \neq 0$ $M = k *_{sm} (0, b, cnj\ b, 0)$ $b \neq 0$
shows $\exists k' \varphi. k' \neq 0 \wedge M = k' *_{sm} (cis\ \varphi, 0, 0, 1) *_{mm} (0, 1, 1, 0)$
 $\langle proof \rangle$

lemma *unitary11-gen-cis-inversion'*:

assumes $k \neq 0$ $M = k *_{sm} (-1, 0, 0, 1) *_{mm} (0, b, cnj\ b, 0)$ $b \neq 0$
shows $\exists k' \varphi. k' \neq 0 \wedge M = k' *_{sm} (cis\ \varphi, 0, 0, 1) *_{mm} (0, 1, 1, 0)$
 $\langle proof \rangle$

lemma *unitary11-gen-cis-inversion-rev*:

assumes $k' \neq 0$ $M = k' *_{sm} (cis\ \varphi, 0, 0, 1) *_{mm} (0, 1, 1, 0)$
shows $\exists k\ a\ b. k \neq 0 \wedge mat-det\ (a, b, cnj\ b, cnj\ a) \neq 0 \wedge M = k *_{sm} (a, b,$
 $cnj\ b, cnj\ a)$
 $\langle proof \rangle$

lemma *unitary11-gen-iff*:

shows *unitary11-gen* $M \longleftrightarrow (\exists k\ a\ b. k \neq 0 \wedge mat-det\ (a, b, cnj\ b, cnj\ a) \neq$
 $0 \wedge M = k *_{sm} (a, b, cnj\ b, cnj\ a))$ (**is** ?lhs = ?rhs)
 $\langle proof \rangle$

lemma *unitary11-iff*:

shows *unitary11* $M \longleftrightarrow$
 $(\exists a\ b\ k. (cmod\ a)^2 > (cmod\ b)^2 \wedge (cmod\ k)^2 = 1 \wedge ((cmod\ a)^2 - (cmod\ b)^2)$
 $\wedge M = k *_{sm} (a, b, cnj\ b, cnj\ a))$ (**is** ?lhs = ?rhs)
 $\langle proof \rangle$

lemma *unitary11-inv*:

assumes $k \neq 0$ $M = k *_{sm} (a, b, cnj\ b, cnj\ a)$ *mat-det* $(a, b, cnj\ b, cnj\ a) \neq 0$
shows $\exists k'\ a'\ b'. k' \neq 0 \wedge mat-inv\ M = k' *_{sm} (a', b', cnj\ b', cnj\ a') \wedge mat-det$
 $(a', b', cnj\ b', cnj\ a') \neq 0$

$\langle \text{proof} \rangle$

lemma *unitary11-comp*:

assumes $k1 \neq 0$ $M1 = k1 *_{sm} (a1, b1, cnj\ b1, cnj\ a1)$ *mat-det* $(a1, b1, cnj\ b1, cnj\ a1) \neq 0$

$k2 \neq 0$ $M2 = k2 *_{sm} (a2, b2, cnj\ b2, cnj\ a2)$ *mat-det* $(a2, b2, cnj\ b2, cnj\ a2) \neq 0$

shows $\exists\ k\ a\ b. k \neq 0 \wedge M1 *_{mm} M2 = k *_{sm} (a, b, cnj\ b, cnj\ a) \wedge \text{mat-det } (a, b, cnj\ b, cnj\ a) \neq 0$

$\langle \text{proof} \rangle$

lemma *unitary11-gen-mat-inv*:

assumes *unitary11-gen* M *mat-det* $M \neq 0$

shows *unitary11-gen* (*mat-inv* M)

$\langle \text{proof} \rangle$

lemma *unitary11-gen-comp*:

assumes *unitary11-gen* $M1$ *mat-det* $M1 \neq 0$ *unitary11-gen* $M2$ *mat-det* $M2 \neq 0$

shows *unitary11-gen* ($M1 *_{mm} M2$)

$\langle \text{proof} \rangle$

lemma *unitary11-sgn-det-orientation*:

assumes $k \neq 0$ *mat-det* $(a, b, cnj\ b, cnj\ a) \neq 0$ $M = k *_{sm} (a, b, cnj\ b, cnj\ a)$

shows $\exists\ k'. \text{sgn } k' = \text{sgn } (\text{Re } (\text{mat-det } (a, b, cnj\ b, cnj\ a))) \wedge \text{congruence } M (1, 0, 0, -1) = \text{cor } k' *_{sm} (1, 0, 0, -1)$

$\langle \text{proof} \rangle$

lemma *unitary11-sgn-det*:

assumes $k \neq 0$ *mat-det* $(a, b, cnj\ b, cnj\ a) \neq 0$ $M = k *_{sm} (a, b, cnj\ b, cnj\ a)$ $M = (A, B, C, D)$

shows $\text{sgn } (\text{Re } (\text{mat-det } (a, b, cnj\ b, cnj\ a))) = (\text{if } b = 0 \text{ then } 1 \text{ else } \text{sgn } (\text{Re } ((A*D)/(B*C)) - 1))$

$\langle \text{proof} \rangle$

lemma *unitary11-orientation*:

assumes *unitary11-gen* M $M = (A, B, C, D)$

shows $\exists\ k'. \text{sgn } k' = \text{sgn } (\text{if } B = 0 \text{ then } 1 \text{ else } \text{sgn } (\text{Re } ((A*D)/(B*C)) - 1)) \wedge \text{congruence } M (1, 0, 0, -1) = \text{cor } k' *_{sm} (1, 0, 0, -1)$

$\langle \text{proof} \rangle$

lemma *unitary11-sgn-det-orientation'*:

assumes *congruence* $M (1, 0, 0, -1) = \text{cor } k' *_{sm} (1, 0, 0, -1)$ $k' \neq 0$

shows $\exists\ a\ b\ k. k \neq 0 \wedge M = k *_{sm} (a, b, cnj\ b, cnj\ a) \wedge \text{sgn } k' = \text{sgn } (\text{Re } (\text{mat-det } (a, b, cnj\ b, cnj\ a)))$

$\langle \text{proof} \rangle$

end

6 Hermitean matrices

theory *HermiteanMatrices*
imports *UnitaryMatrices*
begin

Hermitean matrices

definition *hermitean* :: *complex-mat* \Rightarrow *bool* **where**
hermitean *A* \longleftrightarrow *mat-adj* *A* = *A*

lemma *hermitean* *A* \longleftrightarrow *mat-transpose* *A* = *mat-cnj* *A*
 \langle *proof* \rangle

lemma *hermitean-mat-cnj*: *hermitean* *H* \longleftrightarrow *hermitean* (*mat-cnj* *H*)
 \langle *proof* \rangle

lemma *hermitean-mult-real*:
assumes *hermitean* *H*
shows *hermitean* ((*cor* *k*) *_{sm} *H*)
 \langle *proof* \rangle

lemma *hermitean-congruence*:
assumes *hermitean* *H*
shows *hermitean* (*congruence* *M* *H*)
 \langle *proof* \rangle

lemma *hermitean-elems*:
assumes *hermitean* (*A*, *B*, *C*, *D*)
shows *is-real* *A* *is-real* *D* *B* = *cnj* *C* *cnj* *B* = *C*
 \langle *proof* \rangle

lemma *mat-det-hermitean-real*:
assumes *hermitean* *A*
shows *is-real* (*mat-det* *A*)
 \langle *proof* \rangle

lemma *Re-det-sgn-congruence*:
assumes *hermitean* *H* *mat-det* *M* \neq 0
shows *sgn* (*Re* (*mat-det* (*congruence* *M* *H*))) = *sgn* (*Re* (*mat-det* *H*))
 \langle *proof* \rangle

lemma *det-sgn-congruence*:
assumes *hermitean* *H* *mat-det* *M* \neq 0
shows *sgn* (*mat-det* (*congruence* *M* *H*)) = *sgn* (*mat-det* *H*)
 \langle *proof* \rangle

lemma *bilinear-form-hermitean-commute*:
assumes *hermitean* *H*
shows *bilinear-form* *v1* *v2* *H* = *cnj* (*bilinear-form* *v2* *v1* *H*)

<proof>

lemma *quad-form-hermitean-real:*

assumes *hermitean H*

shows *is-real (quad-form z H)*

<proof>

Eigenvalues, eigenvectors and diagonalization of Hermitean matrices

lemma *hermitean-eigenval-real:*

assumes *hermitean H eigenval k H*

shows *is-real k*

<proof>

lemma *hermitean-distinct-eigenvals:*

assumes *hermitean H*

shows $(\exists k_1 k_2. k_1 \neq k_2 \wedge \text{eigenval } k_1 H \wedge \text{eigenval } k_2 H) \vee \text{mat-diagonal } H$

<proof>

lemma *hermitean-ortho-eigenvecs:*

assumes *hermitean H*

assumes *eigenpair k1 v1 H eigenpair k2 v2 H k1 \neq k2*

shows $\text{vec-cn}j \ v2 \ *_{vv} \ v1 = 0 \ \text{vec-cn}j \ v1 \ *_{vv} \ v2 = 0$

<proof>

lemma *hermitean-diagonalizable:*

assumes *hermitean H*

shows $\exists k1 \ k2 \ M. \text{mat-det } M \neq 0 \wedge \text{unitary } M \wedge \text{congruence } M \ H = (k1, 0, 0, k2) \wedge$
 $\text{is-real } k1 \wedge \text{is-real } k2 \wedge \text{sgn } (\text{Re } k1 * \text{Re } k2) = \text{sgn } (\text{Re } (\text{mat-det } H))$

<proof>

end

7 Elementary complex geometry

theory *ElementaryComplexGeometry*

imports *MoreComplex LinearSystems*

begin

definition *colinear :: complex \Rightarrow complex \Rightarrow complex \Rightarrow bool where*

colinear z1 z2 z3 $\longleftrightarrow z1 = z2 \vee \text{Im } ((z3 - z1)/(z2 - z1)) = 0$

lemma *colinear-ex-real:*

*colinear z1 z2 z3 $\longleftrightarrow (\exists k::\text{real}. z1 = z2 \vee z3 - z1 = \text{complex-of-real } k * (z2 - z1))$*

<proof>

lemma *colinear-sym1*:
 $\text{colinear } z1 \ z2 \ z3 \longleftrightarrow \text{colinear } z1 \ z3 \ z2$
 $\langle \text{proof} \rangle$

lemma *colinear-sym2'*:
assumes $\text{colinear } z1 \ z2 \ z3$
shows $\text{colinear } z2 \ z1 \ z3$
 $\langle \text{proof} \rangle$

lemma *colinear-sym2*:
 $\text{colinear } z1 \ z2 \ z3 \longleftrightarrow \text{colinear } z2 \ z1 \ z3$
 $\langle \text{proof} \rangle$

lemma *colinear-trans1*:
assumes $\text{colinear } z0 \ z2 \ z1 \ \text{colinear } z0 \ z3 \ z1 \ z0 \neq z1$
shows $\text{colinear } z0 \ z2 \ z3$
 $\langle \text{proof} \rangle$

lemma *colinear-det*:
assumes $\neg \text{colinear } z2 \ z3 \ z1$
shows $\det2 \ (z1 - z2) \ (\text{cnj } (z1 - z2)) \ (z2 - z3) \ (\text{cnj } (z2 - z3)) \neq 0$
 $\langle \text{proof} \rangle$

definition *line* :: $\text{complex} \Rightarrow \text{complex} \Rightarrow \text{complex set}$ **where**
 $\text{line } z1 \ z2 = \{z. \text{colinear } z1 \ z2 \ z\}$

lemma *line-points-colinear*:
assumes $z1 \in \text{line } z \ z' \ z2 \in \text{line } z \ z' \ z3 \in \text{line } z \ z' \ z \neq z'$
shows $\text{colinear } z1 \ z2 \ z3$
 $\langle \text{proof} \rangle$

lemma *line-param*:
shows $z1 + \text{complex-of-real } k * (z2 - z1) \in \text{line } z1 \ z2$
 $\langle \text{proof} \rangle$

definition *circle* :: $\text{complex} \Rightarrow \text{real} \Rightarrow \text{complex set}$ **where**
 $\text{circle } \mu \ r = \{z. \text{cmod } (z - \mu) = r\}$

lemma *line-equation*:
assumes $z1 \neq z2 \ \mu = \text{rot90 } (z2 - z1)$
shows $\text{line } z1 \ z2 = \{z. \text{cnj } \mu * z + \mu * \text{cnj } z - (\text{cnj } \mu * z1 + \mu * \text{cnj } z1) = 0\}$
 $\langle \text{proof} \rangle$

lemma *circle-equation*:

assumes $r \geq 0$
shows $\text{circline } \mu \ r = \{z. z * \text{cnj } z - z * \text{cnj } \mu - \text{cnj } z * \mu + \mu * \text{cnj } \mu - \text{complex-of-real } (r * r) = 0\}$
 $\langle \text{proof} \rangle$

definition *circline* **where**
 $\text{circline } A \ BC \ D = \{z. \text{cor } A * z * \text{cnj } z + \text{cnj } BC * z + BC * \text{cnj } z + \text{cor } D = 0\}$

lemma *circline-circle*:
assumes $A \neq 0 \ A * D \leq (\text{cmod } BC)^2$
 $cl = \text{circline } A \ BC \ D$
 $\mu = -BC / \text{complex-of-real } A \ r2 = ((\text{cmod } BC)^2 - A * D) / A^2 \ r = \text{sqrt } r2$
shows $cl = \text{circle } \mu \ r$
 $\langle \text{proof} \rangle$

lemma *circline-ex-circle*:
assumes $A \neq 0 \ A * D \leq (\text{cmod } BC)^2$
 $cl = \text{circline } A \ BC \ D$
shows $\exists \mu \ r. cl = \text{circle } \mu \ r$
 $\langle \text{proof} \rangle$

lemma *circle-circline*:
assumes $cl = \text{circle } \mu \ r \ r \geq 0$
shows $cl = \text{circline } 1 \ (-\mu) ((\text{cmod } \mu)^2 - r^2)$
 $\langle \text{proof} \rangle$

lemma *circle-ex-circline*:
assumes $cl = \text{circle } \mu \ r \ r \geq 0$
shows $\exists A \ BC \ D. A \neq 0 \wedge A * D \leq (\text{cmod } BC)^2 \wedge cl = \text{circline } A \ BC \ D$
 $\langle \text{proof} \rangle$

lemma *circline-line*:
assumes
 $A = 0 \ BC \neq 0$
 $cl = \text{circline } A \ BC \ D$
 $z1 = -\text{cor } D * BC / (2 * BC * \text{cnj } BC)$
 $z2 = z1 + ii * \text{sgn } (\text{if } \arg BC > 0 \text{ then } -BC \text{ else } BC)$
shows
 $cl = \text{line } z1 \ z2$
 $\langle \text{proof} \rangle$

lemma *circline-ex-line*:
assumes
 $A = 0 \ BC \neq 0$
 $cl = \text{circline } A \ BC \ D$

shows $\exists z1\ z2. z1 \neq z2 \wedge cl = line\ z1\ z2$
 $\langle proof \rangle$

lemma *line-ex-circline*:
assumes $cl = line\ z1\ z2\ z1 \neq z2$
shows $\exists BC\ D. BC \neq 0 \wedge cl = circline\ 0\ BC\ D$
 $\langle proof \rangle$

end

theory *Angles*
imports *MoreComplex*
begin

definition *ang-vec* ($()$) **where**
 $[simp]:\ z1\ z2 \equiv \lfloor arg\ z2 - arg\ z1 \rfloor$

definition *ang-vec-c* (c) **where**
 $[simp]:\ c\ z1\ z2 \equiv abs\ (z1\ z2)$

definition *acute-ang* **where**
 $[simp]:\ acute-ang\ \alpha = (if\ \alpha > pi\ /\ 2\ then\ pi - \alpha\ else\ \alpha)$

definition *ang-vec-a* (a) **where**
 $[simp]:\ a\ z1\ z2 \equiv acute-ang\ (c\ z1\ z2)$

lemma *ang-vec-sym*:
assumes $z1\ z2 \neq pi$
shows $z1\ z2 = -\ z2\ z1$
 $\langle proof \rangle$

lemma *ang-vec-sym-pi*:
assumes $z1\ z2 = pi$
shows $z1\ z2 = z2\ z1$
 $\langle proof \rangle$

lemma *ang-vec-c-sym*:
shows $c\ z1\ z2 = c\ z2\ z1$
 $\langle proof \rangle$

lemma *ang-vec-a-sym*:

$a\ z1\ z2 = a\ z2\ z1$

$\langle proof \rangle$

lemma *ang-vec-c-bounded*: $0 \leq c\ z1\ z2 \wedge c\ z1\ z2 \leq \pi$

$\langle proof \rangle$

lemma *ortho-c-scalprod0*:

assumes $z1 \neq 0\ z2 \neq 0$

shows $c\ z1\ z2 = \pi/2 \longleftrightarrow \text{scalprod}\ z1\ z2 = 0$

$\langle proof \rangle$

lemma *ortho-a-scalprod0*:

assumes $z1 \neq 0\ z2 \neq 0$

shows $a\ z1\ z2 = \pi/2 \longleftrightarrow \text{scalprod}\ z1\ z2 = 0$

$\langle proof \rangle$

lemma *canon-ang-plus-pi1*:

assumes $z1\ z2 > 0$

shows $\lfloor z1\ z2 + \pi \rfloor = z1\ z2 - \pi$

$\langle proof \rangle$

lemma *canon-ang-plus-pi2*:

assumes $z1\ z2 \leq 0$

shows $\lfloor z1\ z2 + \pi \rfloor = z1\ z2 + \pi$

$\langle proof \rangle$

lemma *ang-vec-opposite1*:

assumes $z1 \neq 0$

shows $(-z1)\ z2 = \lfloor z1\ z2 - \pi \rfloor$

$\langle proof \rangle$

lemma *ang-vec-opposite2*:

assumes $z2 \neq 0$

shows $z1\ (-z2) = \lfloor z1\ z2 + \pi \rfloor$

$\langle proof \rangle$

lemma *ang-vec-opposite-opposite*:

assumes $z1 \neq 0\ z2 \neq 0$

shows $(-z1)\ (-z2) = z1\ z2$

$\langle proof \rangle$

lemma *ang-vec-a-opposite2*:

$a\ z1\ z2 = a\ z1\ (-z2)$

$\langle proof \rangle$

lemma *ang-vec-a-opposite1*:

$$a \ z1 \ z2 = a \ (-z1) \ z2$$

<proof>

lemma *ang-vec-a-scale1*:

assumes $k \neq 0$

shows $a \ (\text{complex-of-real } k * z1) \ z2 = a \ z1 \ z2$

<proof>

lemma *ang-vec-a-scale2*:

assumes $k \neq 0$

shows $a \ z1 \ (\text{complex-of-real } k * z2) = a \ z1 \ z2$

<proof>

lemma *ang-vec-a-scale*:

assumes $k1 \neq 0 \ k2 \neq 0$

shows $a \ (\text{complex-of-real } k1 * z1) \ (\text{complex-of-real } k2 * z2) = a \ z1 \ z2$

<proof>

lemma *ang-a-cnj-cnj*:

shows $a \ z1 \ z2 = a \ (\text{cnj } z1) \ (\text{cnj } z2)$

<proof>

abbreviation *sgn-bool* **where**

$\text{sgn-bool } p \equiv \text{if } p \text{ then } 1 \text{ else } -1$

definition *circ-tang-vec* :: $\text{complex} \Rightarrow \text{complex} \Rightarrow \text{bool} \Rightarrow \text{complex}$ **where**

$\text{circ-tang-vec } \mu \ E \ p = \text{sgn-bool } p * ii * (E - \mu)$

lemma *circ-tang-vec-ortho*:

$\text{scalprod } (E - \mu) \ (\text{circ-tang-vec } \mu \ E \ p) = 0$

<proof>

lemma *circ-tang-vec-opposite-orient*:

$\text{circ-tang-vec } \mu \ E \ p = - \text{circ-tang-vec } \mu \ E \ (\neg p)$

<proof>

definition *ang-circ* **where**

$\text{ang-circ } E \ \mu1 \ \mu2 \ p1 \ p2 = (\text{circ-tang-vec } \mu1 \ E \ p1) \ (\text{circ-tang-vec } \mu2 \ E \ p2)$

definition *ang-circ-c* **where**

$\text{ang-circ-c } E \ \mu1 \ \mu2 \ p1 \ p2 = c \ (\text{circ-tang-vec } \mu1 \ E \ p1) \ (\text{circ-tang-vec } \mu2 \ E \ p2)$

definition *ang-circ-a* **where**

$\text{ang-circ-a } E \ \mu1 \ \mu2 \ p1 \ p2 = a \ (\text{circ-tang-vec } \mu1 \ E \ p1) \ (\text{circ-tang-vec } \mu2 \ E \ p2)$

lemma *ang-circ-simp*:

assumes $E \neq \mu 1 \ E \neq \mu 2$
shows $\text{ang-circ } E \ \mu 1 \ \mu 2 \ p1 \ p2 = \text{canon-ang } (\arg (E - \mu 2) - \arg (E - \mu 1) + \text{sgn-bool } p1 * \pi / 2 - \text{sgn-bool } p2 * \pi / 2)$
 $\langle \text{proof} \rangle$

lemma *ang-circ-c-simp*:

assumes $E \neq \mu 1 \ E \neq \mu 2$
shows $\text{ang-circ-c } E \ \mu 1 \ \mu 2 \ p1 \ p2 = \text{abs } (\text{canon-ang } (\arg(E - \mu 2) - \arg(E - \mu 1) + (\text{sgn-bool } p1) * \pi / 2 - (\text{sgn-bool } p2) * \pi / 2))$
 $\langle \text{proof} \rangle$

lemma *ang-circ-a-simp*:

assumes $E \neq \mu 1 \ E \neq \mu 2$
shows $\text{ang-circ-a } E \ \mu 1 \ \mu 2 \ p1 \ p2 = \text{acute-ang } (\text{abs } (\text{canon-ang } (\arg(E - \mu 2) - \arg(E - \mu 1) + (\text{sgn-bool } p1) * \pi / 2 - (\text{sgn-bool } p2) * \pi / 2)))$
 $\langle \text{proof} \rangle$

lemma *ang-circ-a-pTrue*:

assumes $E \neq \mu 1 \ E \neq \mu 2$
shows $\text{ang-circ-a } E \ \mu 1 \ \mu 2 \ p1 \ p2 = \text{ang-circ-a } E \ \mu 1 \ \mu 2 \ \text{True } \text{True}$
 $\langle \text{proof} \rangle$

lemma *ang-circ-a-simp1*:

assumes $E \neq \mu 1 \ E \neq \mu 2$
shows $\text{ang-circ-a } E \ \mu 1 \ \mu 2 \ p1 \ p2 = a \ (E - \mu 1) \ (E - \mu 2)$
 $\langle \text{proof} \rangle$

abbreviation *ang-circ-a'* **where**

$\text{ang-circ-a}' \ E \ \mu 1 \ \mu 2 \equiv \text{ang-circ-a } E \ \mu 1 \ \mu 2 \ \text{True } \text{True}$

lemma *ang-circ-a'-simp*:

assumes $z \neq \mu 1 \ z \neq \mu 2$
shows $\text{ang-circ-a}' \ z \ \mu 1 \ \mu 2 = a \ (z - \mu 1) \ (z - \mu 2)$
 $\langle \text{proof} \rangle$

lemma *cos-cmod-scalprod*:

shows $\text{cmod } b * \text{cmod } c * (\cos (b \ c)) = \text{Re } (\text{scalprod } b \ c)$
 $\langle \text{proof} \rangle$

lemma *law-of-cosines*:

shows $(\text{cdist } B \ C)^2 = (\text{cdist } A \ C)^2 + (\text{cdist } A \ B)^2 - 2 * (\text{cdist } A \ C) * (\text{cdist } A \ B) * (\cos ((C - A) \ (B - A)))$
 $\langle \text{proof} \rangle$

declare *ang-vec-c-def* [simp del]

lemma *cos-c-*: $\cos (c \ z1 \ z2) = \cos (z1 \ z2)$

$\langle \text{proof} \rangle$


```

lemma cos-a-c:  $\cos (a \ z1 \ z2) = \text{abs } (\cos (c \ z1 \ z2))$ 
<proof>

end

```

8 Homogeneous coordinates in extended complex plane

```

theory HomogeneousCoordinates
imports MoreComplex Matrices
begin

```

```

typedef homo-coords = {v. v ≠ vec-zero}
<proof>

```

```

lemma obtain-homo-coords:
  fixes x::homo-coords
  obtains A B where
    Rep-homo-coords x = (A, B) A ≠ 0 ∨ B ≠ 0
<proof>

```

```

definition homo-coords-eq :: homo-coords ⇒ homo-coords ⇒ bool (infix ≈ 50)
where
  [simp]:  $z1 \approx z2 \iff$ 
    (let  $z1 = \text{Rep-homo-coords } z1$ ;
      $z2 = \text{Rep-homo-coords } z2$ 
     in  $(\exists k. k \neq (0::\text{complex}) \wedge z2 = k *_{sv} z1)$ )

```

```

lemma homo-coords-eq-reflp:
  reflp homo-coords-eq
<proof>

```

```

lemma homo-coords-eq-symp:
  symp homo-coords-eq
<proof>

```

```

lemma homo-coords-eq-transp:
  transp homo-coords-eq
<proof>

```

```

lemma homo-coords-eq-equivp:
  equivp homo-coords-eq
<proof>

```

```

lemma homo-coords-eq-refl [simp]:
   $z \approx z$ 

```

$\langle \text{proof} \rangle$

lemma *homo-coords-eq-trans*:

assumes $z1 \approx z2 \quad z2 \approx z3$

shows $z1 \approx z3$

$\langle \text{proof} \rangle$

lemma *homo-coords-eq-sym*:

assumes $z1 \approx z2$

shows $z2 \approx z1$

$\langle \text{proof} \rangle$

lemma *homo-coords-eq-mix*:

assumes $\text{Rep-homo-coords } z1 = (z1', z1'') \quad \text{Rep-homo-coords } z2 = (z2', z2'')$

shows $z1 \approx z2 \iff z2' * z1'' = z1' * z2''$

$\langle \text{proof} \rangle$

lemma *[simp]: Rep-homo-coords (Abs-homo-coords (Rep-homo-coords x)) = Rep-homo-coords x*

$\langle \text{proof} \rangle$

Quotient of homogeneous coordinates

quotient-type

complex-homo = homo-coords / homo-coords-eq

$\langle \text{proof} \rangle$

Infinite point

definition *inf-homo-rep* **where** *[simp]: inf-homo-rep = Abs-homo-coords (1, 0)*

lift-definition *inf-homo :: complex-homo (∞_h) is inf-homo-rep*

$\langle \text{proof} \rangle$

lemma *[simp]: Rep-homo-coords (Abs-homo-coords (1, 0)) = (1, 0)*

$\langle \text{proof} \rangle$

lemma *[simp]: Rep-homo-coords inf-homo-rep = (1, 0)*

$\langle \text{proof} \rangle$

lemma *inf-snd-0: $z \approx \text{inf-homo-rep} \iff (\text{let } (z1, z2) = \text{Rep-homo-coords } z \text{ in } z1 \neq 0 \wedge z2 = 0)$*

$\langle \text{proof} \rangle$

lemma *not-inf-snd-not0*:

assumes $\neg z \approx \text{inf-homo-rep}$

shows $\text{let } (z1, z2) = \text{Rep-homo-coords } z \text{ in } z2 \neq 0$

$\langle \text{proof} \rangle$

Zero

definition *zero-homo-rep* **where** *[simp]: zero-homo-rep = Abs-homo-coords (0, 1)*

lift-definition *zero-homo :: complex-homo (0_h) is zero-homo-rep*

$\langle proof \rangle$

lemma $[simp]$: $Rep-homo-coords (Abs-homo-coords (0, 1)) = (0, 1)$
 $\langle proof \rangle$

lemma $[simp]$: $Rep-homo-coords zero-homo-rep = (0, 1)$
 $\langle proof \rangle$

lemma $zero-fst-0$: $z \approx zero-homo-rep \longleftrightarrow (let (z1, z2) = Rep-homo-coords z in z1 = 0 \wedge z2 \neq 0)$
 $\langle proof \rangle$

One

definition $one-homo-rep$ **where** $[simp]$: $one-homo-rep = Abs-homo-coords (1, 1)$
lift-definition $one-homo :: complex-homo (1_h)$ **is** $one-homo-rep$
 $\langle proof \rangle$

lemma $[simp]$: $Rep-homo-coords (Abs-homo-coords (1, 1)) = (1, 1)$
 $\langle proof \rangle$

lemma $[simp]$: $Rep-homo-coords one-homo-rep = (1, 1)$
 $\langle proof \rangle$

lemma $[simp]$: $1_h \neq \infty_h \ 0_h \neq \infty_h \ 0_h \neq 1_h \ 1_h \neq 0_h \ \infty_h \neq 0_h \ \infty_h \neq 1_h$
 $\langle proof \rangle$

definition $ii-homo-rep$ **where** $ii-homo-rep = Abs-homo-coords (ii, 1)$

lift-definition $ii-homo :: complex-homo (ii_h)$ **is** $ii-homo-rep$
 $\langle proof \rangle$

lemma $[simp]$: $Rep-homo-coords (Abs-homo-coords (ii, 1)) = (ii, 1)$
 $\langle proof \rangle$

lemma $[simp]$: $Rep-homo-coords ii-homo-rep = (ii, 1)$
 $\langle proof \rangle$

lemma $ex-3-different-points$:
 fixes $z :: complex-homo$
 shows $\exists z1 z2. z \neq z1 \wedge z1 \neq z2 \wedge z \neq z2$
 $\langle proof \rangle$

Conversion from complex

definition $of-complex-coords$ **where**
 $of-complex-coords z = Abs-homo-coords (z, 1)$

lemma [simp]: *Rep-homo-coords* (*of-complex-coords* z) = (z , 1)
 ⟨proof⟩

lift-definition *of-complex* :: *complex* \Rightarrow *complex-homo* **is** *of-complex-coords*
 ⟨proof⟩

lemma *of-complex-inj*:
 assumes *of-complex* $x = \text{of-complex } y$
 shows $x = y$
 ⟨proof⟩

lemma *of-complex-image-inj*:
 assumes *of-complex* ' $A = \text{of-complex } B$
 shows $A = B$
 ⟨proof⟩

lemma [simp]: *of-complex* $x \neq \infty_h$
 ⟨proof⟩

lemma [simp]: $\infty_h \neq \text{of-complex } x$
 ⟨proof⟩

lemma *inf-homo-or-complex-homo*:
 $z = \infty_h \vee (\exists x. z = \text{of-complex } x)$
 ⟨proof⟩

lemma *zero-of-complex* [simp]: *of-complex* 0 = 0_h
 ⟨proof⟩

lemma *one-of-complex* [simp]: *of-complex* 1 = 1_h
 ⟨proof⟩

lemma
 [simp]: *of-complex* $a = 0_h \longleftrightarrow a = 0$
 ⟨proof⟩

lemma
 [simp]: *of-complex* $a = 1_h \longleftrightarrow a = 1$
 ⟨proof⟩

Coercion to complex

definition *to-complex-homo-coords* :: *homo-coords* \Rightarrow *complex* **where**
to-complex-homo-coords $z = (\text{let } (z1, z2) = \text{Rep-homo-coords } z \text{ in } z1/z2)$

lift-definition *to-complex* :: *complex-homo* \Rightarrow *complex* **is** *to-complex-homo-coords*
 ⟨proof⟩

lemma [simp]: *to-complex* (*of-complex* z) = z

$\langle proof \rangle$

lemma $[simp]$: $z \neq \infty_h \implies (of\text{-}complex\ (to\text{-}complex\ z)) = z$
 $\langle proof \rangle$

Addition

definition $add\text{-}homo\text{-}coords :: homo\text{-}coords \Rightarrow homo\text{-}coords \Rightarrow homo\text{-}coords$ (**infixl** $+_{hc}$ 100) **where**
 $z +_{hc} w = (let\ (z1, z2) = Rep\text{-}homo\text{-}coords\ z;$
 $(w1, w2) = Rep\text{-}homo\text{-}coords\ w\ in$
 $Abs\text{-}homo\text{-}coords\ (z1*w2 + w1*z2, z2*w2))$

lemma $add\text{-}homo\text{-}coords\text{-}Rep$:

assumes $Rep\text{-}homo\text{-}coords\ z = (z1, z2)\ Rep\text{-}homo\text{-}coords\ w = (w1, w2)\ z2 \neq 0$
 $\vee\ w2 \neq 0$
 shows $Rep\text{-}homo\text{-}coords\ (z +_{hc} w) = (z1*w2 + w1*z2, z2*w2)$
 $\langle proof \rangle$

lemma $add\text{-}homo\text{-}coords\text{-}00$:

assumes $Rep\text{-}homo\text{-}coords\ z = (z1, z2)\ Rep\text{-}homo\text{-}coords\ w = (w1, w2)\ z2 = 0$
 $w2 = 0$
 shows $z +_{hc} w = Abs\text{-}homo\text{-}coords\ (0, 0)$
 $\langle proof \rangle$

lemma $add\text{-}coords\text{-}well\text{-}defined\text{-}lemma$:

assumes $x \approx y\ x' \approx y'$
 shows $x +_{hc} x' \approx y +_{hc} y'$
 $\langle proof \rangle$

lift-definition $add\text{-}homo :: complex\text{-}homo \Rightarrow complex\text{-}homo \Rightarrow complex\text{-}homo$ (**infixl** $+_h$ 100) **is** $add\text{-}homo\text{-}coords$
 $\langle proof \rangle$

lemma $add\text{-}homo\text{-}commute$: $x +_h y = y +_h x$
 $\langle proof \rangle$

lemma $of\text{-}complex\text{-}add$: $(of\text{-}complex\ za) +_h (of\text{-}complex\ zb) = of\text{-}complex\ (za + zb)$
 $\langle proof \rangle$

lemma $[simp]$: $(of\text{-}complex\ z) +_h \infty_h = \infty_h$
 $\langle proof \rangle$

lemma $[simp]$: $\infty_h +_h (of\text{-}complex\ z) = \infty_h$
 $\langle proof \rangle$

lemma $add\text{-}homo\text{-}zero\text{-}right$ $[simp]$: $z +_h 0_h = z$
 $\langle proof \rangle$

lemma *add-homo-zero-left* [simp]: $0_h +_h z = z$
 ⟨proof⟩

uminus

definition *uminus-homo-coords* **where**
 $uminus-homo-coords\ z = (let\ (z1, z2) = Rep-homo-coords\ z\ in\ Abs-homo-coords\ (-z1, z2))$

lemma *uminus-homo-coords-Rep* [simp]: $Rep-homo-coords\ (uminus-homo-coords\ z) = (let\ (z1, z2) = Rep-homo-coords\ z\ in\ (-z1, z2))$
 ⟨proof⟩

lift-definition *uminus-homo* :: $complex-homo \Rightarrow complex-homo$ **is** *uminus-homo-coords*
 ⟨proof⟩

lemma *of-complex-uminus* [simp]: $uminus-homo\ (of-complex\ z) = of-complex\ (-z)$
 ⟨proof⟩

Subtraction

definition *minus-homo* :: $complex-homo \Rightarrow complex-homo \Rightarrow complex-homo$ (**infixl** $-_h\ 100$) **where**
 $z1\ -_h\ z2 = z1\ +_h\ (uminus-homo\ z2)$

lemma *minus-homo-coords-Rep*:
assumes $Rep-homo-coords\ z = (z1, z2)\ Rep-homo-coords\ w = (w1, w2)\ z2 \neq 0 \vee w2 \neq 0$
shows $Rep-homo-coords\ (z\ +_{hc}\ (uminus-homo-coords\ w)) = (z1 * w2 - w1 * z2, z2 * w2)$
 ⟨proof⟩

lemma *of-complex-minus*:
 $(of-complex\ z1)\ -_h\ (of-complex\ z2) = of-complex\ (z1 - z2)$
 ⟨proof⟩

lemma [simp]:
assumes $z \neq \infty_h$
shows $z\ -_h\ z = 0_h$
 ⟨proof⟩

lemma *diff-zero-homo*:
assumes $z1\ -_h\ z2 = 0_h\ z1 \neq \infty_h \vee z2 \neq \infty_h$
shows $z1 = z2$
 ⟨proof⟩

Multiplication

definition *mult-homo-coords* :: $homo-coords \Rightarrow homo-coords \Rightarrow homo-coords$ (**infixl** $*_{hc}\ 100$) **where**
 $x\ *_{hc}\ y = (let\ (x1, y1) = Rep-homo-coords\ x;\ (x2, y2) = Rep-homo-coords\ y\ in$

Abs-homo-coords ($x1*x2, y1*y2$)

lemma *mult-homo-coords-Rep*:

assumes *Rep-homo-coords* $x = (Ax, Bx)$ *Rep-homo-coords* $x' = (Ax', Bx')$ ($Bx \neq 0 \vee Ax' \neq 0$) \wedge ($Bx' \neq 0 \vee Ax \neq 0$)
shows *Rep-homo-coords* ($x *_h x'$) = ($Ax*Ax', Bx*Bx'$)
 $\langle proof \rangle$

lemma *mult-homo-coords-00*:

assumes *Rep-homo-coords* $x = (Ax, Bx)$ *Rep-homo-coords* $x' = (Ax', Bx')$ ($Bx = 0 \wedge Ax' = 0$) \vee ($Bx' = 0 \wedge Ax = 0$)
shows $x *_h x' = \text{Abs-homo-coords } (0, 0)$
 $\langle proof \rangle$

lemma *mult-coords-well-defined-lemma*:

assumes $x \approx y$ $x' \approx y'$
shows $x *_h x' \approx y *_h y'$
 $\langle proof \rangle$

lift-definition *mult-homo* :: *complex-homo* \Rightarrow *complex-homo* \Rightarrow *complex-homo*
(infixl $*_h$ **100)** **is** *mult-homo-coords*
 $\langle proof \rangle$

lemma *mult-of-complex*:

shows (*of-complex* $z1$) $*_h$ (*of-complex* $z2$) = *of-complex* ($z1 * z2$)
 $\langle proof \rangle$

lemma *mult-homo-commute*:

shows $z1 *_h z2 = z2 *_h z1$
 $\langle proof \rangle$

lemma *mult-homo-zero-left [simp]*:

assumes $z \neq \infty_h$
shows $0_h *_h z = 0_h$
 $\langle proof \rangle$

lemma *mult-homo-zero-right [simp]*:

assumes $z \neq \infty_h$
shows $z *_h 0_h = 0_h$
 $\langle proof \rangle$

lemma *mult-homo-inf-right [simp]*:

assumes $z \neq 0_h$
shows $z *_h \infty_h = \infty_h$
 $\langle proof \rangle$

lemma *mult-homo-inf-left [simp]*:

assumes $z \neq 0_h$
shows $\infty_h *_h z = \infty_h$

$\langle proof \rangle$

lemma *mult-homo-one-left* [simp]:

shows $1_h *_{\mathbf{h}} z = z$

$\langle proof \rangle$

lemma *mult-homo-one-right* [simp]:

shows $z *_{\mathbf{h}} 1_h = z$

$\langle proof \rangle$

Reciprocal

definition *reciprocal-homo-coords* :: *homo-coords* \Rightarrow *homo-coords* **where**

reciprocal-homo-coords $x = (\text{let } (x1, y1) = \text{Rep-homo-coords } x \text{ in } \text{Abs-homo-coords } (y1, x1))$

lemma *reciprocal-homo-coords-Rep*: *Rep-homo-coords* (*reciprocal-homo-coords* x)

$= (\text{let } (x1, y1) = \text{Rep-homo-coords } x \text{ in } (y1, x1))$

$\langle proof \rangle$

lift-definition *reciprocal-homo* :: *complex-homo* \Rightarrow *complex-homo* **is** *reciprocal-homo-coords*

$\langle proof \rangle$

lemma [simp]: *reciprocal-homo-coords* (*reciprocal-homo-coords* z) = z

$\langle proof \rangle$

lemma [simp]: *reciprocal-homo* (*reciprocal-homo* z) = z

$\langle proof \rangle$

lemma [simp]: *reciprocal-homo* $0_h = \infty_h$

$\langle proof \rangle$

lemma [simp]: *reciprocal-homo* $\infty_h = 0_h$

$\langle proof \rangle$

lemma [simp]: *reciprocal-homo* $1_h = 1_h$

$\langle proof \rangle$

Division

definition *divide-homo* :: *complex-homo* \Rightarrow *complex-homo* \Rightarrow *complex-homo* (**infixl** $:_h$ 100) **where**

$x :_h y = x *_{\mathbf{h}} (\text{reciprocal-homo } y)$

lemma [simp]:

assumes $z \neq 0_h$

shows $z :_h 0_h = \infty_h$

$\langle proof \rangle$

lemma [simp]:

assumes $z \neq \infty_h$

shows $z :_h \infty_h = 0_h$
 $\langle proof \rangle$

lemma $[simp]$: $\infty_h :_h 0_h = \infty_h$
 $\langle proof \rangle$

lemma $[simp]$: $0_h :_h \infty_h = 0_h$
 $\langle proof \rangle$

lemma *divide-homo-one* $[simp]$:
shows $z :_h 1_h = z$
 $\langle proof \rangle$

lemma *of-complex-divide*:
assumes $z2 \neq 0$
shows $(of-complex\ z1) :_h (of-complex\ z2) = of-complex\ (z1 / z2)$
 $\langle proof \rangle$

lemma *divide-homo-coords-Rep* $[simp]$:
assumes $Rep-homo-coords\ z = (z1, z2)\ Rep-homo-coords\ w = (w1, w2)$
 $(z2 \neq 0 \vee w2 \neq 0) \wedge (w1 \neq 0 \vee z1 \neq 0)$
shows $Rep-homo-coords\ (z *_h (reciprocal-homo-coords\ w)) = (z1 * w2, z2 * w1)$
 $\langle proof \rangle$

Conjugate

definition *cnj-homo-coords* **where**
 $cnj-homo-coords\ z = (let\ (z1, z2) = Rep-homo-coords\ z\ in\ Abs-homo-coords\ (cnj\ z1,\ cnj\ z2))$

lemma $[simp]$: $Rep-homo-coords\ (cnj-homo-coords\ z) = vec-cnj\ (Rep-homo-coords\ z)$
 $\langle proof \rangle$

lift-definition *cnj-homo* :: $complex-homo \Rightarrow complex-homo$ **is** *cnj-homo-coords*
 $\langle proof \rangle$

lemma *cnj-homo* $(of-complex\ z) = of-complex\ (cnj\ z)$
 $\langle proof \rangle$

lemma *cnj-homo* $\infty_h = \infty_h$
 $\langle proof \rangle$

lemma *cnj-homo-coords-involution* $[simp]$:
 $cnj-homo-coords\ (cnj-homo-coords\ z) = z$
 $\langle proof \rangle$

lemma *cnj-homo-involution* $[simp]$: $cnj-homo\ (cnj-homo\ z) = z$
 $\langle proof \rangle$

lemma *[simp]*:
 $cnj-homo \circ \infty_h = \infty_h$
 $\langle proof \rangle$

lemma *[simp]*:
 $cnj-homo \circ 0_h = 0_h$
 $\langle proof \rangle$

Inversion

definition *inversion-homo* **where**
 $inversion-homo = cnj-homo \circ reciprocal-homo$

lemma *inversion-homo-sym*:
 $inversion-homo = reciprocal-homo \circ cnj-homo$
 $\langle proof \rangle$

lemma *inversion-homo-involution* *[simp]*: $inversion-homo (inversion-homo z) = z$
 $\langle proof \rangle$

lemma *[simp]*:
 $inversion-homo \circ 0_h = \infty_h$
 $\langle proof \rangle$

lemma *[simp]*:
 $inversion-homo \circ \infty_h = 0_h$
 $\langle proof \rangle$

8.1 Ratio and crossratio

definition *ratio-rep* **where**
 $ratio-rep \ z1 \ z2 \ z3 =$
 $(let \ (z1x, z1y) = Rep-homo-coords \ z1;$
 $\quad (z2x, z2y) = Rep-homo-coords \ z2;$
 $\quad (z3x, z3y) = Rep-homo-coords \ z3 \ in$
 $Abs-homo-coords \ ((z1x*z2y - z2x*z1y)*z3y, (z1x*z3y - z3x*z1y)*z2y))$

lemma *ratio-rep-Rep* *[simp]*:
assumes $(\neg z1 \approx z2 \wedge \neg z3 \approx inf-homo-rep) \vee (\neg z1 \approx z3 \wedge \neg z2 \approx inf-homo-rep)$
shows $Rep-homo-coords \ (ratio-rep \ z1 \ z2 \ z3) = (let \ (z1x, z1y) = Rep-homo-coords \ z1;$
 $\quad (z2x, z2y) = Rep-homo-coords \ z2;$
 $\quad (z3x, z3y) = Rep-homo-coords \ z3 \ in \ ((z1x*z2y - z2x*z1y)*z3y, (z1x*z3y$
 $- z3x*z1y)*z2y))$
 $\langle proof \rangle$

lemma *ratio-rep-Rep'* *[simp]*:
assumes $(z1 \approx z2 \vee z3 \approx inf-homo-rep) \wedge (z1 \approx z3 \vee z2 \approx inf-homo-rep)$
shows $ratio-rep \ z1 \ z2 \ z3 = Abs-homo-coords \ (0, 0)$
 $\langle proof \rangle$

lift-definition *ratio* :: *complex-homo* \Rightarrow *complex-homo* \Rightarrow *complex-homo* \Rightarrow *complex-homo*
is *ratio-rep*
 \langle *proof* \rangle

lemma *ratio-is-ratio*:
assumes $z1 \neq z2 \vee z1 \neq z3 \vee z1 \neq \infty_h \vee z2 \neq \infty_h \vee z3 \neq \infty_h$
shows $\text{ratio } z1 \ z2 \ z3 = (z1 -_h z2) :_h (z1 -_h z3)$
 \langle *proof* \rangle

lemma
assumes $z2 \neq \infty_h \vee z3 \neq \infty_h$
shows $\text{ratio } \infty_h \ z2 \ z3 = 1_h$
 \langle *proof* \rangle

lemma
assumes $z1 \neq \infty_h \vee z3 \neq \infty_h$
shows $\text{ratio } z1 \ \infty_h \ z3 = \infty_h$
 \langle *proof* \rangle

lemma
assumes $z1 \neq \infty_h \vee z2 \neq \infty_h$
shows $\text{ratio } z1 \ z2 \ \infty_h = 0_h$
 \langle *proof* \rangle

lemma
assumes $z1 \neq z2 \vee z1 \neq \infty_h$
shows $\text{ratio } z1 \ z2 \ z1 = \infty_h$
 \langle *proof* \rangle

definition *cross-ratio-rep* **where**
 $\text{cross-ratio-rep } z \ u \ v \ w =$
 $(\text{let } (z', z'') = \text{Rep-homo-coords } z;$
 $(u', u'') = \text{Rep-homo-coords } u;$
 $(v', v'') = \text{Rep-homo-coords } v;$
 $(w', w'') = \text{Rep-homo-coords } w$
 $\text{in Abs-homo-coords } ((z' * u'' - u' * z'') * (v' * w'' - w' * v''),$
 $(z' * w'' - w' * z'') * (v' * u'' - u' * v''))$

lemma *cross-ratio-rep-Rep [simp]*:
assumes $(\neg z1 \approx z2 \wedge \neg z3 \approx z4) \vee (\neg z1 \approx z4 \wedge \neg z2 \approx z3)$
shows $\text{Rep-homo-coords } (\text{cross-ratio-rep } z1 \ z2 \ z3 \ z4) =$
 $(\text{let } (z1', z1'') = \text{Rep-homo-coords } z1;$
 $(z2', z2'') = \text{Rep-homo-coords } z2;$

$(z_3', z_3'') = \text{Rep-homo-coords } z_3;$
 $(z_4', z_4'') = \text{Rep-homo-coords } z_4$
 $\text{in } ((z_1' * z_2'' - z_2' * z_1'') * (z_3' * z_4'' - z_4' * z_3''), (z_1' * z_4'' - z_4' * z_1'') * (z_3' * z_2'' - z_2' * z_3''))$
 $\langle \text{proof} \rangle$

lift-definition $\text{cross-ratio} :: \text{complex-homo} \Rightarrow \text{complex-homo} \Rightarrow \text{complex-homo} \Rightarrow \text{complex-homo} \Rightarrow \text{complex-homo} \Rightarrow \text{cross-ratio-rep}$
 $\langle \text{proof} \rangle$

lemma $\text{cross-ratio } z \ 0_h \ 1_h \ \infty_h = z$
 $\langle \text{proof} \rangle$

lemma cross-ratio-0 :
assumes $z_1 \neq z_2 \ z_1 \neq z_3$
shows $\text{cross-ratio } z_1 \ z_1 \ z_2 \ z_3 = 0_h$
 $\langle \text{proof} \rangle$

lemma cross-ratio-1 :
assumes $z_1 \neq z_2 \ z_2 \neq z_3$
shows $\text{cross-ratio } z_2 \ z_1 \ z_2 \ z_3 = 1_h$
 $\langle \text{proof} \rangle$

lemma cross-ratio-inf :
assumes $z_1 \neq z_3 \ z_2 \neq z_3$
shows $\text{cross-ratio } z_3 \ z_1 \ z_2 \ z_3 = \infty_h$
 $\langle \text{proof} \rangle$

lemma
assumes $(z \neq u \wedge v \neq w) \vee (z \neq w \wedge u \neq v) \ z \neq \infty_h \ u \neq \infty_h \ v \neq \infty_h \ w$
 $\neq \infty_h$
shows $\text{cross-ratio } z \ u \ v \ w = ((z -_h u) *_h (v -_h w)) :_h ((z -_h w) *_h (v -_h u))$
 $\langle \text{proof} \rangle$

8.2 Distance

definition inprod-homo-rep **where**

$\text{inprod-homo-rep } z \ w =$
 $(\text{let } (z_1, z_2) = \text{Rep-homo-coords } z;$
 $(w_1, w_2) = \text{Rep-homo-coords } w$
 $\text{in } \text{vec-cn } (z_1, z_2) *_v (w_1, w_2))$

syntax

$\text{-inprod-homo-rep} :: \text{homo-coords} \Rightarrow \text{homo-coords} \Rightarrow \text{complex } (\langle -, \rangle)$

translations

$\langle z, w \rangle == \text{CONST } \text{inprod-homo-rep } z \ w$

lemma $[\text{simp}]$: $\text{is-real } \langle z, z \rangle$
 $\langle \text{proof} \rangle$

lemma *[simp]*: $\text{Re } \langle z, z \rangle \geq 0$
 $\langle \text{proof} \rangle$

lemma *inprod-homo-bilinear1*:
assumes *Rep-homo-coords* $z' = k *_{sv} \text{Rep-homo-coords } z$
shows $\langle z', w \rangle = \text{cnj } k * \langle z, w \rangle$
 $\langle \text{proof} \rangle$

lemma *inprod-homo-bilinear2*:
assumes *Rep-homo-coords* $w' = k *_{sv} \text{Rep-homo-coords } w$
shows $\langle z, w' \rangle = k * \langle z, w \rangle$
 $\langle \text{proof} \rangle$

definition *norm-homo-rep* **where**
 $\text{norm-homo-rep } z = \text{sqrt } (\text{Re } \langle z, z \rangle)$
syntax
 $\text{-norm-homo-rep} :: \text{homo-coords} \Rightarrow \text{complex } (\langle - \rangle)$
translations
 $\langle z \rangle == \text{CONST norm-homo-rep } z$

lemma
 $\text{norm-homo-rep-square: } \langle z \rangle^2 = \text{Re } (\langle z, z \rangle)$
 $\langle \text{proof} \rangle$

lemma *norm-homo-gt-0*: $\langle z \rangle > 0$
 $\langle \text{proof} \rangle$

lemma *norm-homo-scale*:
assumes *Rep-homo-coords* $z' = k *_{sv} \text{Rep-homo-coords } z$
shows $\langle z' \rangle^2 = \text{Re } (\text{cnj } k * k) * \langle z \rangle^2$
 $\langle \text{proof} \rangle$

definition *dist-homo-rep* **where**
 $\text{dist-homo-rep } z1 \ z2 =$
 $(\text{let } (z1x, z1y) = \text{Rep-homo-coords } z1;$
 $(z2x, z2y) = \text{Rep-homo-coords } z2;$
 $\text{num} = (z1x * z2y - z2x * z1y) * (\text{cnj } z1x * \text{cnj } z2y - \text{cnj } z2x * \text{cnj } z1y);$
 $\text{den} = (z1x * \text{cnj } z1x + z1y * \text{cnj } z1y) * (z2x * \text{cnj } z2x + z2y * \text{cnj } z2y)$
 $\text{in } 2 * \text{sqrt}(\text{Re num} / \text{Re den}))$

lemma *dist-homo-rep-iff*: $\text{dist-homo-rep } z \ w = 2 * \text{sqrt}(1 - (\text{cmod } \langle z, w \rangle)^2 / (\langle z \rangle^2 * \langle w \rangle^2))$
 $\langle \text{proof} \rangle$

lift-definition *dist-homo* :: $\text{complex-homo} \Rightarrow \text{complex-homo} \Rightarrow \text{real}$ **is** *dist-homo-rep*
 $\langle \text{proof} \rangle$

lemma *dist-homo-finite*:
 $\text{dist-homo } (\text{of-complex } z1) (\text{of-complex } z2) = 2 * \text{cmod}(z1 - z2) / (\text{sqrt } (1 + (\text{cmod } z1)^2 + (\text{cmod } z2)^2))$

$z1)^2) * \text{sqrt } (1 + (\text{cmod } z2)^2))$
 $\langle \text{proof} \rangle$

lemma *dist-homo-infinite1*:
 $\text{dist-homo } (\text{of-complex } z1) \propto_h = 2 / \text{sqrt } (1 + (\text{cmod } z1)^2)$
 $\langle \text{proof} \rangle$

lemma *dist-homo-infinite2*:
 $\text{dist-homo } \propto_h (\text{of-complex } z1) = 2 / \text{sqrt } (1 + (\text{cmod } z1)^2)$
 $\langle \text{proof} \rangle$

lemma *dist-homo-rep-zero*:
 $\text{dist-homo-rep } z \ w = 0 \longleftrightarrow (\text{cmod } \langle z, w \rangle)^2 = (\langle z \rangle^2 * \langle w \rangle^2)$
 $\langle \text{proof} \rangle$

lemma *dist-homo-zero1* [simp]: $\text{dist-homo } z \ z = 0$
 $\langle \text{proof} \rangle$

lemma *dist-homo-zero2* [simp]:
assumes $\text{dist-homo } z1 \ z2 = 0$
shows $z1 = z2$
 $\langle \text{proof} \rangle$

lemma *dist-homo-sym* [simp]:
shows $\text{dist-homo } z1 \ z2 = \text{dist-homo } z2 \ z1$
 $\langle \text{proof} \rangle$

Triangle inequality

lemma *dist-homo-triangle-finite*: $\text{cmod}(a - b) / (\text{sqrt } (1 + (\text{cmod } a)^2) * \text{sqrt } (1 + (\text{cmod } b)^2)) \leq \text{cmod } (a - c) / (\text{sqrt } (1 + (\text{cmod } a)^2) * \text{sqrt } (1 + (\text{cmod } c)^2)) + \text{cmod } (c - b) / (\text{sqrt } (1 + (\text{cmod } b)^2) * \text{sqrt } (1 + (\text{cmod } c)^2))$
 $\langle \text{proof} \rangle$

lemma *dist-homo-triangle-infinite1*: $1 / \text{sqrt}(1 + (\text{cmod } b)^2) \leq 1 / \text{sqrt}(1 + (\text{cmod } c)^2) + \text{cmod } (b - c) / (\text{sqrt}(1 + (\text{cmod } b)^2) * \text{sqrt}(1 + (\text{cmod } c)^2))$
 $\langle \text{proof} \rangle$

lemma *dist-homo-triangle-infinite2*:
 $1 / \text{sqrt}(1 + (\text{cmod } a)^2) \leq \text{cmod } (a - c) / (\text{sqrt } (1 + (\text{cmod } a)^2) * \text{sqrt } (1 + (\text{cmod } c)^2)) + 1 / \text{sqrt}(1 + (\text{cmod } c)^2)$
 $\langle \text{proof} \rangle$

lemma *dist-homo-triangle-infinite3*:
 $\text{cmod}(a - b) / (\text{sqrt } (1 + (\text{cmod } a)^2) * \text{sqrt } (1 + (\text{cmod } b)^2)) \leq 1 / \text{sqrt}(1 + (\text{cmod } a)^2) + 1 / \text{sqrt}(1 + (\text{cmod } b)^2)$
 $\langle \text{proof} \rangle$

lemma *dist-homo-triangle*:
shows $\text{dist-homo } A \ B \leq \text{dist-homo } A \ C + \text{dist-homo } C \ B$

$\langle proof \rangle$

instantiation *complex-homo* :: *metric-space*

begin

definition *dist-complex-homo* = *dist-homo*

definition *open-complex-homo* $S = (\forall x \in S. \exists e > 0. \forall y. \text{dist-homo } y \ x < e \longrightarrow y \in S)$

instance

$\langle proof \rangle$

end

end

theory *RiemannSphere*

imports *HomogeneousCoordinates* $\sim\sim$ /src/HOL/Library/Product-Vector

begin

lemma *Lim-within*: $(f \dashrightarrow l) \text{ (at } a \text{ within } S) \longleftrightarrow$

$(\forall e > 0. \exists d > 0. \forall x \in S. 0 < \text{dist } x \ a \wedge \text{dist } x \ a < d \longrightarrow \text{dist } (f \ x) \ l < e)$

$\langle proof \rangle$

lemma *continuous-on-iff*:

continuous-on $s \ f \longleftrightarrow$

$(\forall x \in s. \forall e > 0. \exists d > 0. \forall x' \in s. \text{dist } x' \ x < d \longrightarrow \text{dist } (f \ x') \ (f \ x) < e)$

$\langle proof \rangle$

9 Riemann sphere

typedef *riemann-sphere* = $\{(x::\text{real}, y::\text{real}, z::\text{real}). x*x + y*y + z*z = 1\}$

$\langle proof \rangle$

lemma *sphere-bounds'*:

assumes $x*x + y*y + z*z = (1::\text{real})$

shows $-1 \leq x \wedge x \leq 1$

$\langle proof \rangle$

lemma *sphere-bounds*:

assumes $x*x + y*y + z*z = (1::\text{real})$

shows $-1 \leq x \wedge x \leq 1 \quad -1 \leq y \wedge y \leq 1 \quad -1 \leq z \wedge z \leq 1$

$\langle proof \rangle$

Polar coords parametrization

lemma *sphere-params-on-sphere*:

assumes $x = \cos \alpha * \cos \beta \quad y = \cos \alpha * \sin \beta \quad z = \sin \alpha$

shows $x*x + y*y + z*z = 1$

$\langle proof \rangle$

lemma *sphere-params*:

assumes $x*x + y*y + z*z = 1$
shows $x = \cos (\arcsin z) * \cos (\operatorname{atan2} y x) \wedge y = \cos (\arcsin z) * \sin (\operatorname{atan2} y x) \wedge z = \sin (\arcsin z)$
 $\langle \text{proof} \rangle$

lemma *ex-sphere-params*:

assumes $x*x + y*y + z*z = 1$
shows $\exists \alpha \beta. x = \cos \alpha * \cos \beta \wedge y = \cos \alpha * \sin \beta \wedge z = \sin \alpha \wedge -\pi / 2 \leq \alpha \wedge \alpha \leq \pi / 2 \wedge -\pi \leq \beta \wedge \beta < \pi$
 $\langle \text{proof} \rangle$

Stereographic and inverse stereographic projection

definition *stereographic-coords* :: *riemann-sphere* \Rightarrow *homo-coords* **where**

stereographic-coords $M = (\text{let } (x, y, z) = \text{Rep-riemann-sphere } M \text{ in}$
 (if $(x, y, z) \neq (0, 0, 1)$ then
 Abs-homo-coords (*Complex* $x\ y$, *complex-of-real* $(1 - z)$)
 else
 Abs-homo-coords $(1, 0)$
))

lemma *stereographic-coords-rep*:

Rep-homo-coords (*stereographic-coords* M) = $(\text{let } (x, y, z) = \text{Rep-riemann-sphere } M \text{ in}$
 (if $(x, y, z) \neq (0, 0, 1)$ then
 (*Complex* $x\ y$, *complex-of-real* $(1 - z)$)
 else
 $(1, 0)$
))
 $\langle \text{proof} \rangle$

lift-definition *stereographic* :: *riemann-sphere* \Rightarrow *complex-homo* **is** *stereographic-coords*
 $\langle \text{proof} \rangle$

definition *inv-stereographic-coords* :: *homo-coords* \Rightarrow *riemann-sphere* **where**

inv-stereographic-coords $z =$ (
 let $(z1, z2) = \text{Rep-homo-coords } z$
 in if $z2 = 0$ then
 Abs-riemann-sphere $(0, 0, 1)$
 else
 let $z = z1 / z2$;
 $X = \text{Re } (2*z / (1 + z*cnj\ z))$;
 $Y = \text{Im } (2*z / (1 + z*cnj\ z))$;
 $Z = ((\text{cmod } z)^2 - 1) / (1 + (\text{cmod } z)^2)$
 in *Abs-riemann-sphere* (X, Y, Z))

lift-definition *inv-stereographic* :: *complex-homo* \Rightarrow *riemann-sphere* **is** *inv-stereographic-coords*
 $\langle \text{proof} \rangle$

lemma *one-plus-square-neq-zero* [*simp*]:

fixes $x :: \text{real}$
shows $1 + (\text{cor } x)^2 \neq 0$
 $\langle \text{proof} \rangle$

lemma *Re-stereographic*: $\text{Re } (2 * z / (1 + z * \text{cnj } z)) = 2 * \text{Re } z / (1 + (\text{cmod } z)^2)$
 $\langle \text{proof} \rangle$

lemma *Im-stereographic*: $\text{Im } (2 * z / (1 + z * \text{cnj } z)) = 2 * \text{Im } z / (1 + (\text{cmod } z)^2)$
 $\langle \text{proof} \rangle$

lemma *inv-stereographic-on-sphere*:
assumes $X = \text{Re } (2 * z / (1 + z * \text{cnj } z))$ $Y = \text{Im } (2 * z / (1 + z * \text{cnj } z))$ $Z = ((\text{cmod } z)^2 - 1) / (1 + (\text{cmod } z)^2)$
shows $X * X + Y * Y + Z * Z = 1$
 $\langle \text{proof} \rangle$

lemma *inv-stereographic-coords-Rep*:
 $\text{Rep-riemann-sphere } (\text{inv-stereographic-coords } z) =$
 $(\text{let } (z1, z2) = \text{Rep-homo-coords } z$
 $\text{in if } z2 = 0 \text{ then}$
 $(0, 0, 1)$
 else
 $\text{let } z = z1 / z2;$
 $X = \text{Re } (2 * z / (1 + z * \text{cnj } z));$
 $Y = \text{Im } (2 * z / (1 + z * \text{cnj } z));$
 $Z = ((\text{cmod } z)^2 - 1) / (1 + (\text{cmod } z)^2)$
 $\text{in } (X, Y, Z))$
 $\langle \text{proof} \rangle$

definition [*simp*]: $\text{North} = \text{Abs-riemann-sphere } (0, 0, 1)$

lemma *stereographic-North*: $\text{stereographic } x = \infty_h \longleftrightarrow x = \text{North}$
 $\langle \text{proof} \rangle$

lemma *stereographic-inv-stereographic'*:
assumes
 $z: z = z1 / z2$ **and** $z2 \neq 0$ **and**
 $X: X = \text{Re } (2 * z / (1 + z * \text{cnj } z))$ **and** $Y: Y = \text{Im } (2 * z / (1 + z * \text{cnj } z))$ **and**
 $Z: Z = ((\text{cmod } z)^2 - 1) / (1 + (\text{cmod } z)^2)$
shows $\exists k. k \neq 0 \wedge (\text{Complex } X \ Y, \text{complex-of-real } (1 - Z)) = k *_{sv} (z1, z2)$
 $\langle \text{proof} \rangle$

lemma
stereographic-inv-stereographic:
 $\text{stereographic } (\text{inv-stereographic } z) = z$
 $\langle \text{proof} \rangle$

lemma *bij-stereographic*: *bij stereographic*
 $\langle \text{proof} \rangle$

lemma *inv-stereographic-stereographic*:
 $\text{inv-stereographic} (\text{stereographic } x) = x$
 $\langle \text{proof} \rangle$

lemma *inv-stereographic-is-inv*:
 $\text{inv-stereographic} = \text{inv stereographic}$
 $\langle \text{proof} \rangle$

Circles on the sphere

type-synonym *real-vec-4* = $\text{real} \times \text{real} \times \text{real} \times \text{real}$

fun *mult-sv* :: $\text{real} \Rightarrow \text{real-vec-4} \Rightarrow \text{real-vec-4}$ (**infixl** $*_{sv4}$ 100) **where**
 $k *_{sv4} (a, b, c, d) = (k*a, k*b, k*c, k*d)$

typedef *plane-vec* = $\{(a::\text{real}, b::\text{real}, c::\text{real}, d::\text{real}). a \neq 0 \vee b \neq 0 \vee c \neq 0 \vee d \neq 0\}$
 $\langle \text{proof} \rangle$

definition *plane-vec-eq* **where**
 $\text{plane-vec-eq } v1 \ v2 \longleftrightarrow (\exists k. k \neq 0 \wedge \text{Rep-plane-vec } v2 = k *_{sv4} \text{Rep-plane-vec } v1)$

lemma [*simp*]: $1 *_{sv4} x = x$
 $\langle \text{proof} \rangle$

lemma [*simp*]: $x *_{sv4} (y *_{sv4} v) = (x*y) *_{sv4} v$
 $\langle \text{proof} \rangle$

quotient-type *plane* = *plane-vec* / *plane-vec-eq*
 $\langle \text{proof} \rangle$

definition *on-sphere-circle-rep* **where**
 $\text{on-sphere-circle-rep } \alpha \ A \longleftrightarrow$
 $(\text{let } (X, Y, Z) = \text{Rep-riemann-sphere } A;$
 $(a, b, c, d) = \text{Rep-plane-vec } \alpha$
 $\text{in } a*X + b*Y + c*Z + d = 0)$

lift-definition *on-sphere-circle* :: $\text{plane} \Rightarrow \text{riemann-sphere} \Rightarrow \text{bool}$ **is** *on-sphere-circle-rep*
 $\langle \text{proof} \rangle$

definition *sphere-circle-set* **where**
 $\text{sphere-circle-set } \alpha = \{A. \text{on-sphere-circle } \alpha \ A\}$

Distance on the Riemann sphere

definition *dist-riemann-sphere'* **where**

dist-riemann-sphere' $M1\ M2 =$
 (let ($x1, y1, z1$) = *Rep-riemann-sphere* $M1$;
 ($x2, y2, z2$) = *Rep-riemann-sphere* $M2$
 in norm ($x1 - x2, y1 - y2, z1 - z2$))

lemma *dist-riemann-sphere'-inner*:

(*dist-riemann-sphere'* $M1\ M2$)² = 2 - 2 * inner (*Rep-riemann-sphere* $M1$)
 (*Rep-riemann-sphere* $M2$)
 <proof>

lemma *xxx [simp]*:

Re (2 * $m1$ / (1 + cor ((cmod $m1$)²))) = 2 * Re $m1$ / (1 + (cmod $m1$)²)
 <proof>

lemma *yyy [simp]*:

Im (2 * $m1$ / (1 + cor ((cmod $m1$)²))) = 2 * Im $m1$ / (1 + (cmod $m1$)²)
 <proof>

lemma *dist-riemann-sphere'-ge-0 [simp]*: *dist-riemann-sphere'* $M1\ M2 \geq 0$
 <proof>

lemma *dist-homo-stereographic-finite*:

assumes *stereographic* $M1 = \text{of-complex } m1$ *stereographic* $M2 = \text{of-complex } m2$
shows *dist-riemann-sphere'* $M1\ M2 = 2 * \text{cmod } (m1 - m2) / (\text{sqrt } (1 + (\text{cmod } m1)^2) * \text{sqrt } (1 + (\text{cmod } m2)^2))$
 <proof>

lemma *dist-homo-stereographic-infinite*:

assumes *stereographic* $M1 = \infty_h$ *stereographic* $M2 = \text{of-complex } m2$
shows *dist-riemann-sphere'* $M1\ M2 = 2 / \text{sqrt } (1 + (\text{cmod } m2)^2)$
 <proof>

lemma *dist-riemann-sphere'-sym*: *dist-riemann-sphere'* $M1\ M2 = \text{dist-riemann-sphere' } M2\ M1$
 <proof>

lemma *dist-homo-stereographic*: *dist-riemann-sphere'* $M1\ M2 = \text{dist-homo } (\text{stereographic } M1) (\text{stereographic } M2)$
 <proof>

lemma *dist-homo-stereographic'*:

dist-homo $A\ B = \text{dist-riemann-sphere' } (\text{inv-stereographic } A) (\text{inv-stereographic } B)$
 <proof>

instantiation *riemann-sphere* :: *metric-space*

begin

definition *dist-riemann-sphere* = *dist-riemann-sphere'*

definition *open-riemann-sphere* $S = (\forall x \in S. \exists e > 0. \forall y. \text{dist-riemann-sphere' } y\ x$

```

< e  $\longrightarrow$  y  $\in$  S)
instance
<proof>

end

lemma ex-cos-gt':
  assumes a  $\geq$  0 a < 1  $-pi/2 \leq \alpha \wedge \alpha \leq pi/2$ 
  shows  $\exists \alpha'. -pi/2 \leq \alpha' \wedge \alpha' \leq pi/2 \wedge \alpha' \neq \alpha \wedge \cos (\alpha - \alpha') = a$ 
  <proof>

lemma ex-cos-gt:
  assumes a < 1  $-pi/2 \leq \alpha \wedge \alpha \leq pi/2$ 
  shows  $\exists \alpha'. -pi/2 \leq \alpha' \wedge \alpha' \leq pi/2 \wedge \alpha' \neq \alpha \wedge \cos (\alpha - \alpha') > a$ 
  <proof>

instantiation riemann-sphere :: perfect-space
begin
instance <proof>
end

instantiation complex-homo :: perfect-space
begin
instance <proof>
end

end

lemma continuous-on UNIV stereographic
  <proof>

lemma continuous-on UNIV inv-stereographic
  <proof>

end

```

10 Moebius transformations

```

theory Moebius
imports HomogeneousCoordinates
begin

typedef moebius-mat = {M::complex-mat. mat-det M  $\neq$  0}
  <proof>

definition moebius-mat-eq where
  [simp]: moebius-mat-eq A B  $\longleftrightarrow (\exists k::complex. k \neq 0 \wedge Rep\text{-}moebius\text{-}mat\ B = k *_{sm} (Rep\text{-}moebius\text{-}mat\ A))$ 

```

lemma *[simp]: moebius-mat-eq x x*
 $\langle \text{proof} \rangle$

quotient-type *moebius = moebius-mat / moebius-mat-eq*
 $\langle \text{proof} \rangle$

definition *mk-moebius-rep where*
 $\text{mk-moebius-rep } a \ b \ c \ d = \text{Abs-moebius-mat } (a, b, c, d)$

lift-definition *mk-moebius :: complex \Rightarrow complex \Rightarrow complex \Rightarrow complex \Rightarrow moebius is mk-moebius-rep*
 $\langle \text{proof} \rangle$

lemma *mk-moebius-rep-Rep:*
assumes *mat-det (a, b, c, d) $\neq 0$*
shows *Rep-moebius-mat (mk-moebius-rep a b c d) = (a, b, c, d)*
 $\langle \text{proof} \rangle$

lemma *ex-mk-moebius:*
shows $\exists \ a \ b \ c \ d. M = \text{mk-moebius } a \ b \ c \ d \wedge \text{mat-det } (a, b, c, d) \neq 0$
 $\langle \text{proof} \rangle$

10.1 Action on points

definition *moebius-pt-rep :: moebius-mat \Rightarrow homo-coords \Rightarrow homo-coords where*

moebius-pt-rep M z =
 $(\text{let } z = \text{Rep-homo-coords } z;$
 $\quad M = \text{Rep-moebius-mat } M$
 $\text{in Abs-homo-coords } (M *_{mv} z))$

lemma *[simp]: Rep-homo-coords (Abs-homo-coords (Rep-moebius-mat M $*_{mv}$ Rep-homo-coords x)) = Rep-moebius-mat M $*_{mv}$ Rep-homo-coords x*
 $\langle \text{proof} \rangle$

lemma *[simp]: Rep-homo-coords (moebius-pt-rep M z) = Rep-moebius-mat M $*_{mv}$ Rep-homo-coords z*
 $\langle \text{proof} \rangle$

lift-definition *moebius-pt :: moebius \Rightarrow complex-homo \Rightarrow complex-homo is moebius-pt-rep*
 $\langle \text{proof} \rangle$

lemma *bij-moebius-pt:*
shows *bij (moebius-pt M)*
 $\langle \text{proof} \rangle$

definition *is-moebius where*
 $\text{is-moebius } f \longleftrightarrow (\exists \ M. f = \text{moebius-pt } M)$

Bilinear and linear expressions

lemma *moebius-bilinear*:
assumes *mat-det* (*a*, *b*, *c*, *d*) $\neq 0$
shows *moebius-pt* (*mk-moebius* *a b c d*) *z* =
 (*if* *z* $\neq \infty_h$ *then*
 ((*of-complex* *a*) \ast_h *z* $+_h$ (*of-complex* *b*)) $:_h$
 ((*of-complex* *c*) \ast_h *z* $+_h$ (*of-complex* *d*))
 else
 (*of-complex* *a*) $:_h$
 (*of-complex* *c*)
<proof>

10.2 Moebius group

definition *moebius-inv-rep* **where**
moebius-inv-rep *M* =
 (*let* *M* = *Rep-moebius-mat* *M*
 in *Abs-moebius-mat* (*mat-inv* *M*))

lemma [*simp*]: *Rep-moebius-mat* (*Abs-moebius-mat* (*mat-inv* (*Rep-moebius-mat* *M*))) = *mat-inv* (*Rep-moebius-mat* *M*)
<proof>

lemma [*simp*]: *Rep-moebius-mat* (*moebius-inv-rep* *M*) = *mat-inv* (*Rep-moebius-mat* *M*)
<proof>

lift-definition *moebius-inv* :: *moebius* \Rightarrow *moebius* **is** *moebius-inv-rep*
<proof>

lemma *moebius-inv*: *moebius-pt* (*moebius-inv* *M*) = *inv* (*moebius-pt* *M*)
<proof>

lemma *is-moebius-inv*:
assumes *is-moebius* *m*
shows *is-moebius* (*inv* *m*)
<proof>

definition *moebius-comp-rep* **where**
moebius-comp-rep *M1* *M2* =
 (*let* *M1* = *Rep-moebius-mat* *M1*;
 M2 = *Rep-moebius-mat* *M2* *in*
 Abs-moebius-mat (*M1* \ast_{mm} *M2*))

lemma [*simp*]: *Rep-moebius-mat* (*Abs-moebius-mat* ((*Rep-moebius-mat* *M1*) \ast_{mm} (*Rep-moebius-mat* *M2*))) = (*Rep-moebius-mat* *M1*) \ast_{mm} (*Rep-moebius-mat* *M2*)
<proof>

lemma [*simp*]: *Rep-moebius-mat* (*moebius-comp-rep* *M1* *M2*) = (*Rep-moebius-mat*

$M1) *_{mm} (Rep\text{-}moebius\text{-}mat\ M2)$
 $\langle proof \rangle$

lift-definition $moebius\text{-}comp :: moebius \Rightarrow moebius \Rightarrow moebius$ **is** $moebius\text{-}comp\text{-}rep$
 $\langle proof \rangle$

lemma $moebius\text{-}comp$: $moebius\text{-}pt\ M1 \circ moebius\text{-}pt\ M2 = moebius\text{-}pt\ (moebius\text{-}comp\ M1\ M2)$
 $\langle proof \rangle$

lemma $is\text{-}moebius\text{-}comp$:
assumes $is\text{-}moebius\ m1\ is\text{-}moebius\ m2$
shows $is\text{-}moebius\ (m1 \circ m2)$
 $\langle proof \rangle$

definition $[simp]$: $id\text{-}moebius\text{-}rep = Abs\text{-}moebius\text{-}mat\ eye$

lift-definition $id\text{-}moebius :: moebius$ **is** $id\text{-}moebius\text{-}rep$
 $\langle proof \rangle$

lemma $[simp]$: $Rep\text{-}moebius\text{-}mat\ (Abs\text{-}moebius\text{-}mat\ (1, 0, 0, 1)) = eye$
 $\langle proof \rangle$

lemma $[simp]$: $Rep\text{-}moebius\text{-}mat\ (id\text{-}moebius\text{-}rep) = eye$
 $\langle proof \rangle$

lemma $moebius\text{-}pt\ id\text{-}moebius = id$
 $\langle proof \rangle$

instantiation $moebius :: group\text{-}add$
begin

definition $plus\text{-}moebius :: moebius \Rightarrow moebius \Rightarrow moebius$ **where**
 $[simp]$: $plus\text{-}moebius = moebius\text{-}comp$

definition $uminus\text{-}moebius :: moebius \Rightarrow moebius$ **where**
 $[simp]$: $uminus\text{-}moebius = moebius\text{-}inv$

definition $zero\text{-}moebius :: moebius$ **where**
 $[simp]$: $zero\text{-}moebius = id\text{-}moebius$

definition $minus\text{-}moebius :: moebius \Rightarrow moebius \Rightarrow moebius$ **where**
 $[simp]$: $minus\text{-}moebius\ A\ B = A + (-B)$

instance $\langle proof \rangle$
end

lemma $[simp]$: $moebius\text{-}comp\ (moebius\text{-}inv\ M)\ M = id\text{-}moebius$
 $\langle proof \rangle$

lemma [simp]: *moebius-comp* *M* (*moebius-inv* *M*) = *id-moebius*
 <proof>

lemma *moebius-pt-moebius-id* [simp]: *moebius-pt* (*id-moebius*) = *id*
 <proof>

lemma [simp]: *moebius-pt* (*moebius-inv* *M*) (*moebius-pt* *M* *z*) = *z*
 <proof>

lemma *moebius-pt-invert*:
 assumes *w* = *moebius-pt* *M* *z*
 shows *z* = *moebius-pt* (*moebius-inv* *M*) *w*
 <proof>

10.3 Special kinds of Moebius transformations

Reciprocal ($1/z$) as a moebius transformation

definition *reciprocal-moebius* :: *moebius* **where**
reciprocal-moebius = *mk-moebius* 0 1 1 0

lemma [simp]: *Rep-moebius-mat* (*Abs-moebius-mat* (0, 1, 1, 0)) = (0, 1, 1, 0)
 <proof>

lemma [simp]: *Rep-moebius-mat* (*mk-moebius-rep* 0 1 1 0) = (0, 1, 1, 0)
 <proof>

lemma [simp]: *Rep-homo-coords* (*reciprocal-homo-coords* *z*) = (let (*x*, *y*) = *Rep-homo-coords* *z* in (*y*, *x*))
 <proof>

lemma *reciprocal-moebius*:
reciprocal-homo = *moebius-pt* *reciprocal-moebius*
 <proof>

lemma *reciprocal-moebius-inv* [simp]:
moebius-inv *reciprocal-moebius* = *reciprocal-moebius*
 <proof>

lemma *reciprocal-homo-only-0-to-inf*:
 assumes *reciprocal-homo* *z* = ∞_h
 shows *z* = 0_h
 <proof>

lemma *reciprocal-homo-only-inf-to-0*:
 assumes *reciprocal-homo* *z* = 0_h
 shows *z* = ∞_h
 <proof>

Euclidean similarity as a Moebius transform

definition *similarity-moebius* :: *complex* \Rightarrow *complex* \Rightarrow *moebius* **where**
similarity-moebius *a b* = *mk-moebius* *a b 0 1*

lemma *moebius-similarity-linear*:

assumes *a* $\neq 0$

shows *moebius-pt* (*similarity-moebius* *a b*) *z* = (*of-complex* *a*) \ast_h *z* $+_h$ (*of-complex* *b*)

\langle *proof* \rangle

lemma *moebius-similarity'*:

assumes *a* $\neq 0$

shows *moebius-pt* (*similarity-moebius* *a b*) = (λ *z*. (*of-complex* *a*) \ast_h *z* $+_h$ (*of-complex* *b*))

\langle *proof* \rangle

lemma *is-moebius-similarity'*:

assumes *a* $\neq 0_h$ *a* $\neq \infty_h$ *b* $\neq \infty_h$

shows (λ *z*. *a* \ast_h *z* $+_h$ *b*) = *moebius-pt* (*similarity-moebius* (*to-complex* *a*) (*to-complex* *b*))

\langle *proof* \rangle

lemma *is-moebius-similarity*:

assumes *a* $\neq 0_h$ *a* $\neq \infty_h$ *b* $\neq \infty_h$

shows *is-moebius* (λ *z*. *a* \ast_h *z* $+_h$ *b*)

\langle *proof* \rangle

lemma *similarity-moebius-comp*:

assumes *a* $\neq 0$ *c* $\neq 0$

shows *similarity-moebius* *a b* $+$ *similarity-moebius* *c d* = *similarity-moebius* (*a* \ast *c*) (*a* \ast *d* $+$ *b*)

\langle *proof* \rangle

lemma *similarity-moebius-inv*:

assumes *a* $\neq 0$

shows $-$ *similarity-moebius* *a b* = *similarity-moebius* (*1*/*a*) ($-$ *b*/*a*)

\langle *proof* \rangle

lemma *similarity-moebius-id*: *id-moebius* = *similarity-moebius* *1 0*

\langle *proof* \rangle

lemma *similarity-inf-fixed*:

assumes *a* $\neq 0$

shows *moebius-pt* (*similarity-moebius* *a b*) ∞_h = ∞_h

\langle *proof* \rangle

lemma *similarity-only-inf-to-inf*:

assumes *a* $\neq 0$ *moebius-pt* (*similarity-moebius* *a b*) *z* = ∞_h

shows *z* = ∞_h

\langle *proof* \rangle

lemma *inf-fixed-similarity*:

assumes *moebius-pt* $M \infty_h = \infty_h$

shows $\exists a b. a \neq 0 \wedge M = \text{similarity-moebius } a b$

<proof>

Translation

definition *translation-moebius* **where**

translation-moebius $v = \text{similarity-moebius } 1 v$

lemma *translation-moebius-comp*:

$(\text{translation-moebius } v1) + (\text{translation-moebius } v2) = \text{translation-moebius } (v1 + v2)$

<proof>

lemma *translation-moebius-zero*:

translation-moebius $0 = \text{id-moebius}$

<proof>

lemma *moebius-translation-inv*:

$-(\text{translation-moebius } v1) = \text{translation-moebius } (-v1)$

<proof>

lemma *moebius-pt-translation* [simp]: *moebius-pt* $(\text{translation-moebius } v)$ (*of-complex* z) = *of-complex* $(v + z)$

<proof>

Rotation

definition *rotation-moebius* **where**

rotation-moebius $\varphi = \text{similarity-moebius } (\text{cis } \varphi) 0$

lemma *rotation-moebius-comp*:

$(\text{rotation-moebius } \varphi1) + (\text{rotation-moebius } \varphi2) = \text{rotation-moebius } (\varphi1 + \varphi2)$

<proof>

lemma *rotation-moebius-zero*:

rotation-moebius $0 = \text{id-moebius}$

<proof>

lemma *rotation-moebius-inverse*:

$-(\text{rotation-moebius } \varphi) = \text{rotation-moebius } (-\varphi)$

<proof>

lemma *moebius-pt-rotation* [simp]: *moebius-pt* $(\text{rotation-moebius } \varphi)$ (*of-complex* z) = *of-complex* $(\text{cis } \varphi * z)$

<proof>

Dilatation

definition *dilatation-moebius* **where**

dilatation-moebius a = similarity-moebius (cor a) 0

lemma *dilatation-moebius-comp:*

assumes $a1 > 0$ $a2 > 0$

shows $(\text{dilatation-moebius } a1) + (\text{dilatation-moebius } a2) = \text{dilatation-moebius } (a1 * a2)$
 $\langle \text{proof} \rangle$

lemma *dilatation-moebius-zero:*

dilatation-moebius 1 = id-moebius

$\langle \text{proof} \rangle$

lemma *dilatation-moebius-inverse:*

assumes $a > 0$

shows $-(\text{dilatation-moebius } a) = \text{dilatation-moebius } (1/a)$
 $\langle \text{proof} \rangle$

lemma *moebius-pt-dilatation [simp]:* $a \neq 0 \implies \text{moebius-pt } (\text{dilatation-moebius } a)$
 $(\text{of-complex } z) = \text{of-complex } (\text{cor } a * z)$
 $\langle \text{proof} \rangle$

rotation-dilatation-moebius

definition *rotation-dilatation-moebius where*

rotation-dilatation-moebius a = similarity-moebius a 0

lemma *rot-dil:*

assumes $a \neq 0$

shows $\text{rotation-dilatation-moebius } a = \text{rotation-moebius } (\arg a) + \text{dilatation-moebius } (\text{cmod } a)$
 $\langle \text{proof} \rangle$

10.4 Decomposition

lemma *similarity-decomposition:*

assumes $a \neq 0$

shows $\text{similarity-moebius } a \ b = (\text{translation-moebius } b) + (\text{rotation-moebius } (\arg a)) + (\text{dilatation-moebius } (\text{cmod } a))$
 $\langle \text{proof} \rangle$

lemma *moebius-decomposition:*

assumes $c \neq 0$ $a*d - b*c \neq 0$

shows $\text{mk-moebius } a \ b \ c \ d =$

$\text{translation-moebius } (a/c) +$
 $\text{rotation-dilatation-moebius } ((b*c - a*d)/(c*c)) +$
 $\text{reciprocal-moebius } +$
 $\text{translation-moebius } (d/c)$

$\langle \text{proof} \rangle$

lemma *wlog-moebius-decomposition:*

assumes
trans: $\bigwedge v. P \text{ (translation-moebius } v)$ **and** *rot*: $\bigwedge \alpha. P \text{ (rotation-moebius } \alpha)$ **and**
dil: $\bigwedge k. P \text{ (dilatation-moebius } k)$ **and** *recip*: $P \text{ (reciprocal-moebius)}$ **and**
comp: $\bigwedge M1\ M2. \llbracket P\ M1; P\ M2 \rrbracket \implies P\ (M1 + M2)$
shows $P\ M$
 $\langle \text{proof} \rangle$

10.5 Cross ratio and moebius existence

lemma *is-moebius-cross-ratio*:
assumes $z1 \neq z2\ z2 \neq z3\ z1 \neq z3$
shows *is-moebius* $(\lambda z. \text{cross-ratio } z\ z1\ z2\ z3)$
 $\langle \text{proof} \rangle$

lemma *ex-moebius-01inf*:
assumes $z1 \neq z2\ z1 \neq z3\ z2 \neq z3$
shows $\exists M. ((\text{moebius-pt } M\ z1 = 0_h) \wedge (\text{moebius-pt } M\ z2 = 1_h) \wedge (\text{moebius-pt } M\ z3 = \infty_h))$
 $\langle \text{proof} \rangle$

lemma *ex-moebius*:
assumes $z1 \neq z2\ z1 \neq z3\ z2 \neq z3\ w1 \neq w2\ w1 \neq w3\ w2 \neq w3$
shows $\exists M. ((\text{moebius-pt } M\ z1 = w1) \wedge (\text{moebius-pt } M\ z2 = w2) \wedge (\text{moebius-pt } M\ z3 = w3))$
 $\langle \text{proof} \rangle$

lemma *ex-moebius-1*:
shows $\exists M. \text{moebius-pt } M\ z1 = w1$
 $\langle \text{proof} \rangle$

lemma *wlog-moebius-01inf*:
fixes $M::\text{moebius}$
assumes $P\ 0_h\ 1_h\ \infty_h\ z1 \neq z2\ z2 \neq z3\ z1 \neq z3$
 $\bigwedge M\ a\ b\ c. P\ a\ b\ c \implies P\ (\text{moebius-pt } M\ a)\ (\text{moebius-pt } M\ b)\ (\text{moebius-pt } M\ c)$
shows $P\ z1\ z2\ z3$
 $\langle \text{proof} \rangle$

10.6 Fixed points and moebius uniqueness

lemma *three-fixed-points-01inf*:
assumes $\text{moebius-pt } M\ 0_h = 0_h\ \text{moebius-pt } M\ 1_h = 1_h\ \text{moebius-pt } M\ \infty_h = \infty_h$
shows $M = \text{id-moebius}$
 $\langle \text{proof} \rangle$

lemma *three-fixed-points*:
assumes $z1 \neq z2\ z1 \neq z3\ z2 \neq z3$
assumes $\text{moebius-pt } M\ z1 = z1\ \text{moebius-pt } M\ z2 = z2\ \text{moebius-pt } M\ z3 = z3$
shows $M = \text{id-moebius}$
 $\langle \text{proof} \rangle$

lemma *unique-moebius-three-points*:

assumes $z1 \neq z2 \ z1 \neq z3 \ z2 \neq z3$

assumes $\text{moebius-pt } M1 \ z1 = w1 \ \text{moebius-pt } M1 \ z2 = w2 \ \text{moebius-pt } M1 \ z3 = w3$

$\text{moebius-pt } M2 \ z1 = w1 \ \text{moebius-pt } M2 \ z2 = w2 \ \text{moebius-pt } M2 \ z3 = w3$

shows $M1 = M2$

<proof>

lemma *ex-unique-moebius-three-points*:

assumes $z1 \neq z2 \ z1 \neq z3 \ z2 \neq z3 \ w1 \neq w2 \ w1 \neq w3 \ w2 \neq w3$

shows $\exists! M. ((\text{moebius-pt } M \ z1 = w1) \wedge (\text{moebius-pt } M \ z2 = w2) \wedge (\text{moebius-pt } M \ z3 = w3))$

<proof>

lemma *ex-unique-moebius-three-points-fun*:

assumes $z1 \neq z2 \ z1 \neq z3 \ z2 \neq z3 \ w1 \neq w2 \ w1 \neq w3 \ w2 \neq w3$

shows $\exists! f. \text{is-moebius } f \wedge (f \ z1 = w1) \wedge (f \ z2 = w2) \wedge (f \ z3 = w3)$

<proof>

lemma *is-cross-ratio-01inf*:

assumes $z1 \neq z2 \ z1 \neq z3 \ z2 \neq z3 \ \text{is-moebius } f$

assumes $f \ z1 = 0_h \ f \ z2 = 1_h \ f \ z3 = \infty_h$

shows $f = (\lambda z. \text{cross-ratio } z \ z1 \ z2 \ z3)$

<proof>

lemma *moebius-preserve-cross-ratio*:

assumes $z1 \neq z2 \ z1 \neq z3 \ z2 \neq z3$

shows $\text{cross-ratio } z \ z1 \ z2 \ z3 = \text{cross-ratio } (\text{moebius-pt } M \ z) (\text{moebius-pt } M \ z1) (\text{moebius-pt } M \ z2) (\text{moebius-pt } M \ z3)$

<proof>

lemma *fixed-points-0inf'*:

assumes $\text{moebius-pt } M \ 0_h = 0_h \ \text{moebius-pt } M \ \infty_h = \infty_h$

shows $\exists k::\text{complex-homo. } (k \neq 0_h \wedge k \neq \infty_h) \wedge (\forall z. \text{moebius-pt } M \ z = k *_{\text{h}} z)$

<proof>

lemma *fixed-points-0inf*:

assumes $\text{moebius-pt } M \ 0_h = 0_h \ \text{moebius-pt } M \ \infty_h = \infty_h$

shows $\exists k::\text{complex-homo. } (k \neq 0_h \wedge k \neq \infty_h) \wedge \text{moebius-pt } M = (\lambda z. k *_{\text{h}} z)$

<proof>

10.7 Pole

definition *is-pole* where

$\text{is-pole } M \ z \longleftrightarrow \text{moebius-pt } M \ z = \infty_h$

lemma *ex1-pole*:

$\exists! z. \text{is-pole } M z$

$\langle \text{proof} \rangle$

definition *pole* **where** $\text{pole } M = (\text{THE } z. \text{is-pole } M z)$

lemma *pole-mk-moebius*:

assumes *is-pole* (*mk-moebius* $a b c d$) $z c \neq 0$ $a*d - b*c \neq 0$

shows $z = \text{of-complex } (-d/c)$

$\langle \text{proof} \rangle$

lemma *pole-similarity*:

assumes *is-pole* (*similarity-moebius* $a b$) $z a \neq 0$

shows $z = \infty_h$

$\langle \text{proof} \rangle$

10.8 Antihomographies

definition *is-antihomography* **where**

$\text{is-antihomography } f \longleftrightarrow (\exists f'. \text{is-moebius } f' \wedge f = f' \circ \text{cnj-homo})$

lemma *is-antihomography inversion-homo*

$\langle \text{proof} \rangle$

10.9 Classification

lemma *similarity-scale-1*:

assumes $k \neq 0$

shows *similarity* ($k *_{sm} I$) $M = \text{similarity } I M$

$\langle \text{proof} \rangle$

lemma *similarity-scale-2*:

shows *similarity* $I (k *_{sm} M) = k *_{sm} (\text{similarity } I M)$

$\langle \text{proof} \rangle$

lemma [*simp*]: *mat-trace* ($k *_{sm} M$) = $k * \text{mat-trace } M$

$\langle \text{proof} \rangle$

definition *moebius-mb-rep* **where**

$\text{moebius-mb-rep } I M = \text{Abs-moebius-mat } (\text{similarity } (\text{Rep-moebius-mat } I) (\text{Rep-moebius-mat } M))$

lemma *moebius-mb-rep-Rep* [*simp*]:

$\text{Rep-moebius-mat } (\text{moebius-mb-rep } I M) = \text{similarity } (\text{Rep-moebius-mat } I) (\text{Rep-moebius-mat } M)$

$\langle \text{proof} \rangle$

lift-definition *moebius-mb* :: *moebius* \Rightarrow *moebius* \Rightarrow *moebius* **is** *moebius-mb-rep*

$\langle \text{proof} \rangle$

definition *similarity-invar-rep* **where**

similarity-invar-rep $M =$
 (*let* $M = \text{Rep-moebius-mat } M$
 in $(\text{mat-trace } M)^2 / \text{mat-det } M - 4$)

lift-definition *similarity-invar* $:: \text{moebius} \Rightarrow \text{complex}$ **is** *similarity-invar-rep*
 $\langle \text{proof} \rangle$

lemma

similarity-invar (*moebius-mb* I M) = *similarity-invar* M
 $\langle \text{proof} \rangle$

definition *similar* **where**

similar $M1$ $M2 \longleftrightarrow (\exists I. \text{moebius-mb } I \text{ } M1 = M2)$

lemma [*simp*]: *similarity eye* $M = M$
 $\langle \text{proof} \rangle$

lemma [*simp*]: *similarity* $(1, 0, 0, 1)$ $M = M$
 $\langle \text{proof} \rangle$

lemma *similarity-comp*:

assumes *mat-det* $I1 \neq 0$ *mat-det* $I2 \neq 0$

shows *similarity* $I1$ (*similarity* $I2$ M) = *similarity* ($I2 *_{mm} I1$) M
 $\langle \text{proof} \rangle$

lemma *similarity-inv*:

assumes *similarity* I $M1 = M2$ *mat-det* $I \neq 0$

shows *similarity* (*mat-inv* I) $M2 = M1$
 $\langle \text{proof} \rangle$

lemma *similar-refl* [*simp*]: *similar* M M
 $\langle \text{proof} \rangle$

lemma *similar-sym*:

assumes *similar* $M1$ $M2$

shows *similar* $M2$ $M1$
 $\langle \text{proof} \rangle$

lemma *similar-trans*:

assumes *similar* $M1$ $M2$ *similar* $M2$ $M3$

shows *similar* $M1$ $M3$
 $\langle \text{proof} \rangle$

end

11 Circline

```

theory Circline
imports Moebius HermiteanMatrices ElementaryComplexGeometry RiemannSphere
Angles
begin

```

11.1 Circline definition

```

typedef circline-mat = {H. hermitean H ∧ H ≠ mat-zero}
⟨proof⟩

```

```

lemma circline-mat-mult-sm-Rep [simp]:
  assumes k ≠ 0
  shows Rep-circline-mat (Abs-circline-mat ((cor k) *sm (Rep-circline-mat H)))
= (cor k) *sm (Rep-circline-mat H)
⟨proof⟩

```

```

definition circline-mat-eq where
  [simp]: circline-mat-eq A B ⟷ (∃ k::real. k ≠ 0 ∧ Rep-circline-mat B =
complex-of-real k *sm (Rep-circline-mat A))

```

```

lemma [simp]: circline-mat-eq H H
⟨proof⟩

```

```

quotient-type circline = circline-mat / circline-mat-eq
⟨proof⟩

```

Circline with specified matrix

```

definition mk-circline-rep where
  mk-circline-rep A B C D = Abs-circline-mat (A, B, C, D)

```

```

lift-definition mk-circline :: complex ⇒ complex ⇒ complex ⇒ complex ⇒ cir-
cline is mk-circline-rep
⟨proof⟩

```

```

lemma ex-mk-circline:
  shows ∃ A B C D. H = mk-circline A B C D ∧ hermitean (A, B, C, D) ∧ (A, B, C, D) ≠ mat-zero
⟨proof⟩

```

circline type

```

definition circline-type-rep where
  circline-type-rep H = sgn (Re (mat-det (Rep-circline-mat H)))

```

```

lift-definition circline-type :: circline ⇒ real is circline-type-rep
⟨proof⟩

```

```

lemma circline-type: circline-type H = -1 ∨ circline-type H = 0 ∨ circline-type
H = 1

```


$\langle \text{proof} \rangle$

on-circline, *circline-set*

definition *on-circline-rep* **where**

on-circline-rep $H \ z \longleftrightarrow$
 $(\text{let } z = \text{Rep-homo-coords } z;$
 $\quad H = \text{Rep-circline-mat } H$
 $\text{in quad-form } z \ H = 0)$

lift-definition *on-circline* :: *circline* \Rightarrow *complex-homo* \Rightarrow *bool* **is** *on-circline-rep*

$\langle \text{proof} \rangle$

definition *circline-set* :: *circline* \Rightarrow *complex-homo set* **where**

circline-set $H = \{z. \text{on-circline } H \ z\}$

Circlines through 0 and inf

definition *circline-A0-rep* **where**

circline-A0-rep $H \longleftrightarrow$
 $(\text{let } (A, B, C, D) = \text{Rep-circline-mat } H \text{ in } A = 0)$

lift-definition *circline-A0* :: *circline* \Rightarrow *bool* **is** *circline-A0-rep*

$\langle \text{proof} \rangle$

definition *circline-D0-rep* **where**

circline-D0-rep $H \longleftrightarrow$
 $(\text{let } (A, B, C, D) = \text{Rep-circline-mat } H \text{ in } D = 0)$

abbreviation *is-line* **where**

is-line $H \equiv \text{circline-A0 } H$

abbreviation *is-circle* **where**

is-circle $H \equiv \neg \text{circline-A0 } H$

lift-definition *circline-D0* :: *circline* \Rightarrow *bool* **is** *circline-D0-rep*

$\langle \text{proof} \rangle$

lemma *inf-on-circline-rep*: *on-circline-rep* H *inf-homo-rep* \longleftrightarrow *circline-A0-rep* H

$\langle \text{proof} \rangle$

lemma

inf-in-circline-set: $\infty_h \in \text{circline-set } H \longleftrightarrow \text{is-line } H$

$\langle \text{proof} \rangle$

lemma *zero-on-circline-rep*: *on-circline-rep* H *zero-homo-rep* \longleftrightarrow *circline-D0-rep* H

$\langle \text{proof} \rangle$

lemma *zero-in-circline-set*: $0_h \in \text{circline-set } H \longleftrightarrow \text{circline-D0 } H$

$\langle proof \rangle$

Connection with circlines in classic complex plane

lemma *classic-circline*:

assumes $H = mk\text{-circline } A \ B \ C \ D \ hermitean \ (A, B, C, D) \wedge (A, B, C, D) \neq mat\text{-zero}$

shows $circline\text{-set } H - \{\infty_h\} = of\text{-complex } ' \ circline \ (Re \ A) \ B \ (Re \ D)$
 $\langle proof \rangle$

definition *mk-circle-rep* **where**

$mk\text{-circle-rep } a \ r = Abs\text{-circline-mat } (1, -a, -cnj \ a, a * cnj \ a - cor \ r * cor \ r)$

lift-definition *mk-circle* $:: complex \Rightarrow real \Rightarrow circline$ **is** *mk-circle-rep*
 $\langle proof \rangle$

lemma *mk-circle-rep-Rep*

$[simp]: Rep\text{-circline-mat } (mk\text{-circle-rep } a \ r) = (1, -a, -cnj \ a, a * cnj \ a - cor \ r * cor \ r)$
 $\langle proof \rangle$

lemma *is-circle-mk-circle*: $is\text{-circle } (mk\text{-circle } a \ r)$

$\langle proof \rangle$

lemma

assumes $r \geq 0$

shows $circline\text{-set } (mk\text{-circle } a \ r) = of\text{-complex } ' \ \{z. cmod \ (z - a) = r\}$
 $\langle proof \rangle$

definition *mk-line-rep* **where** $mk\text{-line-rep } z1 \ z2 =$

$(let \ B = ii * (z2 - z1) \ in \ Abs\text{-circline-mat } (0, B, cnj \ B, -cnj\text{-mix } B \ z1))$

lift-definition *mk-line* $:: complex \Rightarrow complex \Rightarrow circline$ **is** *mk-line-rep*
 $\langle proof \rangle$

lemma *mk-line-rep-Rep* $[simp]:$

assumes $z1 \neq z2$

shows $Rep\text{-circline-mat } (mk\text{-line-rep } z1 \ z2) =$
 $(let \ B = ii * (z2 - z1) \ in \ (0, B, cnj \ B, -cnj\text{-mix } B \ z1))$
 $\langle proof \rangle$

lemma *circline-line'*:

assumes $z1 \neq z2$

shows $circline \ 0 \ (i * (z2 - z1)) \ (Re \ (- \ cnj\text{-mix } (i * (z2 - z1)) \ z1)) = line \ z1 \ z2$
 $\langle proof \rangle$

lemma

assumes $z1 \neq z2$

shows $circline\text{-set } (mk\text{-line } z1 \ z2) - \{\infty_h\} = of\text{-complex } ' \ line \ z1 \ z2$
 $\langle proof \rangle$

definition *euclidean-circle-rep* **where**

euclidean-circle-rep $H = (\text{let } (A, B, C, D) = \text{Rep-circline-mat } H \text{ in } (-B/A, \text{sqrt}(\text{Re}((B*C - A*D)/(A*A))))$

lift-definition *euclidean-circle* $:: \text{circline} \Rightarrow \text{complex} \times \text{real}$ **is** *euclidean-circle-rep*
 $\langle \text{proof} \rangle$

lemma *classic-circle*:

assumes *is-circle* H $(a, r) = \text{euclidean-circle } H$ *circline-type* $H \leq 0$

shows *circline-set* $H = \text{of-complex } \text{'circle } a \ r$

$\langle \text{proof} \rangle$

definition

euclidean-line-rep $H =$
 $(\text{let } (A, B, C, D) = \text{Rep-circline-mat } H;$
 $z1 = -(D*B)/(2*B*C);$
 $z2 = z1 + ii*\text{sgn}(\text{if } \arg B > 0 \text{ then } -B \text{ else } B)$
 $\text{in } (z1, z2))$

lift-definition *euclidean-line* $:: \text{circline} \Rightarrow \text{complex} \times \text{complex}$ **is** *euclidean-line-rep*
 $\langle \text{proof} \rangle$

lemma *classic-line*:

assumes *is-line* H $(z1, z2) = \text{euclidean-line } H$ *circline-type* $H < 0$

shows *circline-set* $H = \{\infty_h\} = \text{of-complex } \text{'line } z1 \ z2$

$\langle \text{proof} \rangle$

11.2 Connections with circles on the Riemann sphere

definition *inv-stereographic-circline-rep* **where**

inv-stereographic-circline-rep $H =$
 $(\text{let } (A, B, C, D) = \text{Rep-circline-mat } H \text{ in}$
 $\text{Abs-plane-vec } (\text{Re } (B+C), \text{Re}(ii*(C-B)), \text{Re}(A-D), \text{Re}(D+A)))$

lemma *inv-stereographic-circline-rep-Rep* [simp]:

Rep-plane-vec (*inv-stereographic-circline-rep* H) =
 $(\text{let } (A, B, C, D) = \text{Rep-circline-mat } H \text{ in } (\text{Re } (B+C), \text{Re}(ii*(C-B)),$
 $\text{Re}(A-D), \text{Re}(D+A)))$

$\langle \text{proof} \rangle$

lift-definition *inv-stereographic-circline* $:: \text{circline} \Rightarrow \text{plane}$ **is** *inv-stereographic-circline-rep*
 $\langle \text{proof} \rangle$

definition *stereographic-circline-rep* **where**

stereographic-circline-rep $\alpha =$
 $(\text{let } (a, b, c, d) = \text{Rep-plane-vec } \alpha \text{ in}$
 $\text{Abs-circline-mat } (\text{cor } ((c+d)/2), ((\text{cor } a + ii* \text{cor } b)/2), ((\text{cor } a - ii* \text{cor } b)/2), \text{cor } ((d-c)/2)))$

lemma *stereographic-circline-rep-Rep*:

Rep-circline-mat (stereographic-circline-rep α) =
(let (a, b, c, d) = Rep-plane-vec α in
(cor ((c+d)/2) , ((cor a+ii cor b)/2), ((cor a-ii*cor b)/2), cor*
((d-c)/2)))
 <proof>

lift-definition *stereographic-circline :: plane \Rightarrow circline is stereographic-circline-rep*
 <proof>

lemma *stereographic-circline-inv-stereographic-circline*:

stereographic-circline \circ inv-stereographic-circline = id
 <proof>

lemma [simp]: *Im (z / 2) = Im z / 2*

<proof>

lemma [simp]: *(Complex a b) / 2 = Complex (a/2) (b/2)*

<proof>

lemma [simp]: *Complex 2 0 = 2*

<proof>

lemma *inv-stereographic-circline-stereographic-circline*:

inv-stereographic-circline \circ stereographic-circline = id
 <proof>

lemma *stereographic-sphere-circle-set''*:

on-sphere-circle (inv-stereographic-circline H) z \longleftrightarrow on-circline H (stereographic
z)
 <proof>

lemma *stereographic-sphere-circle-set'*:

stereographic ' sphere-circle-set (inv-stereographic-circline H) = circline-set H
 <proof>

lemma *stereographic-sphere-circle-set*:

shows *stereographic ' sphere-circle-set H = circline-set (stereographic-circline H)*
 <proof>

lemma *bij stereographic-circline*

<proof>

lemma *bij inv-stereographic-circline*

<proof>

11.3 Some special circlines

Unit circle

definition *unit-circle-rep* **where**

[simp]: *unit-circle-rep* = *Abs-circline-mat* (1, 0, 0, -1)

lemma [simp]: *Rep-circline-mat* (*Abs-circline-mat* (1, 0, 0, -1)) = (1, 0, 0, -1)
 <proof>

lemma [simp]: *Rep-circline-mat unit-circle-rep* = (1, 0, 0, -1)
 <proof>

lift-definition *unit-circle* :: *circline* **is** *unit-circle-rep*
 <proof>

lemma *one-on-unit-circle*: $1_h \in \text{circline-set } \text{unit-circle}$
 <proof>

x-axis

definition *x-axis-rep* **where** *x-axis-rep* = *Abs-circline-mat* (0, *ii*, -*ii*, 0)

lift-definition *x-axis* :: *circline* **is** *x-axis-rep*
 <proof>

lemma [simp]: *Rep-circline-mat* (*Abs-circline-mat* (0, *ii*, -*ii*, 0)) = (0, *ii*, -*ii*, 0)
 <proof>

lemma [simp]: *Rep-circline-mat x-axis-rep* = (0, *ii*, -*ii*, 0)
 <proof>

lemma [simp]: $0_h \in \text{circline-set } x\text{-axis} \quad 1_h \in \text{circline-set } x\text{-axis} \quad \infty_h \in \text{circline-set } x\text{-axis}$
 <proof>

Point 0_h as a circline

definition *circline-point-0h-rep* **where** *circline-point-0h-rep* = *Abs-circline-mat* (1, 0, 0, 0)

lift-definition *circline-point-0h* :: *circline* **is** *circline-point-0h-rep*
 <proof>

lemma [simp]: *Rep-circline-mat* (*Abs-circline-mat* (1, 0, 0, 0)) = (1, 0, 0, 0)
 <proof>

lemma [simp]: *Rep-circline-mat circline-point-0h-rep* = (1, 0, 0, 0)
 <proof>

imaginary unit circle

definition *imag-unit-circle-rep* **where**

[simp]: *imag-unit-circle-rep* = *Abs-circline-mat* (1, 0, 0, 1)

lemma [simp]: *Rep-circline-mat* (*Abs-circline-mat* (1, 0, 0, 1)) = (1, 0, 0, 1)

$\langle \text{proof} \rangle$

lemma [simp]: $\text{Rep-circline-mat } \text{imag-unit-circle-rep} = (1, 0, 0, 1)$
 $\langle \text{proof} \rangle$

lift-definition $\text{imag-unit-circle} :: \text{circline} \text{ is } \text{imag-unit-circle-rep}$
 $\langle \text{proof} \rangle$

11.4 Moebius action on circlines

definition $\text{moebius-circline-rep} :: \text{moebius-mat} \Rightarrow \text{circline-mat} \Rightarrow \text{circline-mat}$ **where**

$\text{moebius-circline-rep } M \ H =$
 $(\text{let } M = \text{Rep-moebius-mat } M;$
 $H = \text{Rep-circline-mat } H$
 $\text{in } \text{Abs-circline-mat } (\text{congruence } (\text{mat-inv } M) \ H))$

lemma [simp]: $\text{Rep-circline-mat } (\text{Abs-circline-mat } (\text{congruence } (\text{mat-inv } (\text{Rep-moebius-mat } M)) (\text{Rep-circline-mat } H))) = \text{congruence } (\text{mat-inv } (\text{Rep-moebius-mat } M)) (\text{Rep-circline-mat } H)$
 $\langle \text{proof} \rangle$

lemma $\text{moebius-circline-rep-Rep}$ [simp]: $\text{Rep-circline-mat } (\text{moebius-circline-rep } M \ H) = \text{congruence } (\text{mat-inv } (\text{Rep-moebius-mat } M)) (\text{Rep-circline-mat } H)$
 $\langle \text{proof} \rangle$

lift-definition $\text{moebius-circline} :: \text{moebius} \Rightarrow \text{circline} \Rightarrow \text{circline}$ **is** $\text{moebius-circline-rep}$
 $\langle \text{proof} \rangle$

lemma $\text{moebius-preserve-circline-type}$:
 shows $\text{circline-type } (\text{moebius-circline } M \ H) = \text{circline-type } H$
 $\langle \text{proof} \rangle$

lemma $\text{moebius-circline-rep}$:
 shows $\text{moebius-pt-rep } M \ ' \ \{z. \text{on-circline-rep } H \ z\} = \{z. \text{on-circline-rep } (\text{moebius-circline-rep } M \ H) \ z\}$
 $\langle \text{proof} \rangle$

lemma $\text{moebius-circline-set}$:
 shows $\text{moebius-pt } M \ ' \ \text{circline-set } H = \text{circline-set } (\text{moebius-circline } M \ H)$ (**is**
 $?lhs = ?rhs$)
 $\langle \text{proof} \rangle$

lemma
 $\text{inj-moebius-circline}: \text{inj } (\text{moebius-circline } M)$
 $\langle \text{proof} \rangle$

lemma [simp]:
 $\text{moebius-circline } \text{id-moebius } H = H$

$\langle proof \rangle$

lemma *moebius-circline-comp*:

$moebius-circline\ M1\ (moebius-circline\ M2\ H) = moebius-circline\ (moebius-comp\ M1\ M2)\ H$

$\langle proof \rangle$

lemma *moebius-circline-comp-inv* [simp]:

$moebius-circline\ (moebius-inv\ M)\ (moebius-circline\ M\ H) = H$

$\langle proof \rangle$

lemma *moebius-circline-comp-inv'* [simp]:

$moebius-circline\ M\ (moebius-circline\ (moebius-inv\ M)\ H) = H$

$\langle proof \rangle$

lemma

moebius-circline-set-mem:

$moebius-pt\ M\ z \in circline-set\ (moebius-circline\ M\ H) \longleftrightarrow z \in circline-set\ H$

$\langle proof \rangle$

11.5 Conjugation, reciprocation and inversion of circlines

Conjugation of circlines

definition *circline-cnj-rep* **where**

$circline-cnj-rep\ H = Abs-circline-mat\ (mat-cnj\ (Rep-circline-mat\ H))$

lemma [simp]: $Rep-circline-mat\ (Abs-circline-mat\ (mat-cnj\ (Rep-circline-mat\ H))) = mat-cnj\ (Rep-circline-mat\ H)$

$\langle proof \rangle$

lemma [simp]: $Rep-circline-mat\ (circline-cnj-rep\ H) = mat-cnj\ (Rep-circline-mat\ H)$

$\langle proof \rangle$

lift-definition *circline-cnj* :: *circline* \Rightarrow *circline* **is** *circline-cnj-rep*

$\langle proof \rangle$

lemma *cnj-homo-circline-set'*:

shows *cnj-homo* ‘ $circline-set\ H \subseteq circline-set\ (circline-cnj\ H)$

$\langle proof \rangle$

lemma [simp]: $circline-cnj\ (circline-cnj\ H) = H$

$\langle proof \rangle$

lemma *cnj-homo-circline-set*:

shows *cnj-homo* ‘ $circline-set\ H = circline-set\ (circline-cnj\ H)$ (**is** ?lhs = ?rhs)

$\langle proof \rangle$

Reciprocal and inversion of circlines

definition *circline-swap-AD-rep* **where**

circline-swap-AD-rep $H =$
 $(\text{let } (A, B, C, D) = \text{Rep-circline-mat } H$
 $\text{in Abs-circline-mat } (D, B, C, A))$

lemma

shows $[\text{simp}]$: $\text{Rep-circline-mat } (\text{circline-swap-AD-rep } H) = (\text{let } (A, B, C, D)$
 $= \text{Rep-circline-mat } H \text{ in } (D, B, C, A))$
 $\langle \text{proof} \rangle$

lift-definition *circline-swap-AD* :: *circline* \Rightarrow *circline* **is** *circline-swap-AD-rep*
 $\langle \text{proof} \rangle$

lemma *reciprocal-circline-set*:

shows *reciprocal-homo* ‘ *circline-set* $H = \text{circline-set } ((\text{circline-cnj} \circ \text{circline-swap-AD})$
 $H)$
 $\langle \text{proof} \rangle$

lemma *inversion-circline-set*:

shows *inversion-homo* ‘ *circline-set* $H = \text{circline-set } (\text{circline-swap-AD } H)$
 $\langle \text{proof} \rangle$

11.6 Circline uniqueness

11.6.1 Zero type circline uniqueness

lemma *unique-circline-type-zero-0h*’:

shows $(\text{circline-type } \text{circline-point-0h} = 0 \wedge 0_h \in \text{circline-set } \text{circline-point-0h})$
 \wedge
 $(\forall H. \text{circline-type } H = 0 \wedge 0_h \in \text{circline-set } H \longrightarrow H = \text{circline-point-0h})$
 $\langle \text{proof} \rangle$

lemma *unique-circline-type-zero-0h*:

shows $\exists! H. \text{circline-type } H = 0 \wedge 0_h \in \text{circline-set } H$
 $\langle \text{proof} \rangle$

lemma *unique-circline-type-zero*:

shows $\exists! H. \text{circline-type } H = 0 \wedge z \in \text{circline-set } H$
 $\langle \text{proof} \rangle$

11.6.2 Negative type circline uniqueness

lemma *unique-circline-01inf*’:

$0_h \in \text{circline-set } x\text{-axis} \wedge 1_h \in \text{circline-set } x\text{-axis} \wedge \infty_h \in \text{circline-set } x\text{-axis} \wedge$
 $(\forall H. 0_h \in \text{circline-set } H \wedge 1_h \in \text{circline-set } H \wedge \infty_h \in \text{circline-set } H \longrightarrow H$
 $= x\text{-axis})$
 $\langle \text{proof} \rangle$

lemma *unique-circline-set*:

assumes $A \neq B \wedge A \neq C \wedge B \neq C$

shows $\exists! H. A \in \text{circline-set } H \wedge B \in \text{circline-set } H \wedge C \in \text{circline-set } H$
 $\langle \text{proof} \rangle$

11.7 Circline set cardinality

11.7.1 Diagonal circlines

definition *circline-diag-rep* where

$$\text{circline-diag-rep } H \longleftrightarrow \text{mat-diagonal } (\text{Rep-circline-mat } H)$$

lemma *[simp]*: $\text{mat-diagonal } H \longleftrightarrow (\text{let } (A, B, C, D) = H \text{ in } B = 0 \wedge C = 0)$
 $\langle \text{proof} \rangle$

lift-definition *circline-diag* :: *circline* \Rightarrow bool **is** *circline-diag-rep*
 $\langle \text{proof} \rangle$

lemma *det-zero-trace-zero*:

assumes $\text{mat-det } A = 0 \text{ mat-trace } A = (0::\text{complex}) \text{ hermitean } A$

shows $A = \text{mat-zero}$

$\langle \text{proof} \rangle$

lemma *circline-diagonalize*:

shows $\exists M H'. \text{moebius-circline } M H = H' \wedge \text{circline-diag } H'$

$\langle \text{proof} \rangle$

lemma *wlog-circline-diag*:

assumes $\bigwedge H. \text{circline-diag } H \Longrightarrow P H$

$\bigwedge M H. P H \Longrightarrow P (\text{moebius-circline } M H)$

shows $P H$

$\langle \text{proof} \rangle$

11.7.2 Zero type circline set cardinality

lemma *circline-type-zero-card-eq1-0h*:

assumes $\text{circline-type } H = 0 \ 0_h \in \text{circline-set } H$

shows $\text{circline-set } H = \{0_h\}$

$\langle \text{proof} \rangle$

lemma *bij-image-singleton*:

$\llbracket f ' A = \{b\}; f a = b; \text{bij } f \rrbracket \Longrightarrow A = \{a\}$

$\langle \text{proof} \rangle$

lemma *circline-type-zero-card-eq1*:

assumes $\text{circline-type } H = 0$

shows $\exists z. \text{circline-set } H = \{z\}$

$\langle \text{proof} \rangle$

11.7.3 Negative type circline set cardinality

lemma *quad-form-diagonal-iff*:

assumes $k1 \neq 0$ *is-real* $k1$ *is-real* $k2$ $\text{Re } k1 * \text{Re } k2 < 0$
shows $\text{quad-form } (z1, 1) (k1, 0, 0, k2) = 0 \longleftrightarrow (\exists \varphi. z1 = \text{rcis } (\text{sqrt } (\text{Re } (-k2 / k1))) \varphi)$
 $\langle \text{proof} \rangle$

lemma *circline-type-neg-card-gt3-diag*:

assumes *circline-type* $H < 0$ *circline-diag* H
shows $\exists A B C. A \neq B \wedge A \neq C \wedge B \neq C \wedge \{A, B, C\} \subseteq \text{circline-set } H$
 $\langle \text{proof} \rangle$

lemma *circline-type-neg-card-gt3*:

assumes *circline-type* $H < 0$
shows $\exists A B C. A \neq B \wedge A \neq C \wedge B \neq C \wedge \{A, B, C\} \subseteq \text{circline-set } H$
 $\langle \text{proof} \rangle$

11.7.4 Positive type circline set cardinality

lemma *circline-type-pos-card-eq0-diag*:

assumes *circline-diag* H *circline-type* $H > 0$
shows $\text{circline-set } H = \{\}$
 $\langle \text{proof} \rangle$

lemma *circline-type-pos-card-eq0*:

assumes *circline-type* $H > 0$
shows $\text{circline-set } H = \{\}$
 $\langle \text{proof} \rangle$

11.7.5 Cardinality determines type

lemma *card-eq1-circline-type-zero*:

assumes $\exists z. \text{circline-set } H = \{z\}$
shows *circline-type* $H = 0$
 $\langle \text{proof} \rangle$

11.7.6 Circline set is injective

lemma *inj-circline-set*:

assumes $\text{circline-set } H = \text{circline-set } H'$ $\text{circline-set } H \neq \{\}$
shows $H = H'$
 $\langle \text{proof} \rangle$

11.8 Symmetric points wrt. circline

definition *circline-symmetric-rep* **where**

$\text{circline-symmetric-rep } z1 \ z2 \ H \longleftrightarrow$
 $(\text{let } z1 = \text{Rep-homo-coords } z1;$
 $\quad z2 = \text{Rep-homo-coords } z2;$

$H = \text{Rep-circline-mat } H \text{ in}$
 $\text{bilinear-form } z1 \ z2 \ H = 0)$

lift-definition *circline-symmetric* :: *complex-homo* \Rightarrow *complex-homo* \Rightarrow *circline*
 \Rightarrow *bool* **is** *circline-symmetric-rep*
 $\langle \text{proof} \rangle$

lemma *symmetry-principle*:
assumes *circline-symmetric* *z1 z2 H*
shows *circline-symmetric* (*moebius-pt* *M z1*) (*moebius-pt* *M z2*) (*moebius-circline*
M H)
 $\langle \text{proof} \rangle$

Symmetry wrt. *unit-circle*

lemma *circline-symmetric-0inf-disc*: *circline-symmetric* $0_h \propto_h$ *unit-circle*
 $\langle \text{proof} \rangle$

lemma *circline-symmetric-inv-homo-disc*: *circline-symmetric* *a* (*inversion-homo*
a) *unit-circle*
 $\langle \text{proof} \rangle$

lemma *circline-symmetric-inv-homo-disc'*:
assumes *circline-symmetric* *a a' unit-circle*
shows $a' = \text{inversion-homo } a$
 $\langle \text{proof} \rangle$

11.9 Oriented circlines; discs

definition *ocircline-mat-eq* **where**
 $[\text{simp}]: \text{ocircline-mat-eq } A \ B \longleftrightarrow (\exists \ k::\text{real}. \ k > 0 \wedge \text{Rep-circline-mat } B =$
 $\text{complex-of-real } k *_{sm} (\text{Rep-circline-mat } A))$

lemma $[\text{simp}]: \text{ocircline-mat-eq } H \ H$
 $\langle \text{proof} \rangle$

quotient-type *ocircline* = *circline-mat* / *ocircline-mat-eq*
 $\langle \text{proof} \rangle$

lift-definition *on-ocircline* :: *ocircline* \Rightarrow *complex-homo* \Rightarrow *bool* **is** *on-circline-rep*
 $\langle \text{proof} \rangle$

definition *ocircline-set* :: *ocircline* \Rightarrow *complex-homo set* **where**
ocircline-set *H* = $\{z. \text{on-ocircline } H \ z\}$

disc and disc complement

definition *in-ocircline-rep* **where**
in-ocircline-rep *H z* \longleftrightarrow
 $(\text{let } z = \text{Rep-homo-coords } z;$
 $\quad H = \text{Rep-circline-mat } H$

$$\text{in } \text{Re } (\text{quad-form } z \ H) < 0)$$

lift-definition *in-ocircline* :: *ocircline* \Rightarrow *complex-homo* \Rightarrow *bool* **is** *in-ocircline-rep*
 $\langle \text{proof} \rangle$

definition *disc* **where**
 $\text{disc } H = \{z. \text{in-ocircline } H \ z\}$

definition *out-ocircline-rep* **where**
 $\text{out-ocircline-rep } H \ z \longleftrightarrow$
 $(\text{let } z = \text{Rep-homo-coords } z;$
 $H = \text{Rep-circline-mat } H$
 $\text{in } \text{Re } (\text{quad-form } z \ H) > 0)$

lift-definition *out-ocircline* :: *ocircline* \Rightarrow *complex-homo* \Rightarrow *bool* **is** *out-ocircline-rep*
 $\langle \text{proof} \rangle$

definition *disc-compl* **where**
 $\text{disc-compl } H = \{z. \text{out-ocircline } H \ z\}$

lemma *in-on-out*: $\text{in-ocircline } H \ z \vee \text{on-ocircline } H \ z \vee \text{out-ocircline } H \ z$
 $\langle \text{proof} \rangle$

lemma $\text{disc } H \cup \text{disc-compl } H \cup \text{ocircline-set } H = \text{UNIV}$
 $\langle \text{proof} \rangle$

lemma
 $\text{disc-inter-disc-compl}: \text{disc } H \cap \text{disc-compl } H = \{\}$
 $\langle \text{proof} \rangle$

lemma
 $\text{disc-inter-ocircline-set}: \text{disc } H \cap \text{ocircline-set } H = \{\}$
 $\langle \text{proof} \rangle$

lemma
 $\text{disc-compl-inter-ocircline-set}: \text{disc-compl } H \cap \text{ocircline-set } H = \{\}$
 $\langle \text{proof} \rangle$

Opposite orientation

definition *opposite-ocircline-rep* **where**
 $\text{opposite-ocircline-rep } H =$
 $(\text{let } H = \text{Rep-circline-mat } H \text{ in}$
 $\text{Abs-circline-mat } (-1 \ *_{sm} \ H))$

lemma *circline-mat-mult-m1* [*simp*]: $\text{Rep-circline-mat } (\text{Abs-circline-mat } (-1 \ *_{sm} \ \text{Rep-circline-mat } H)) = (-1 \ *_{sm} \ \text{Rep-circline-mat } H)$
 $\langle \text{proof} \rangle$

lemma [*simp*]: $\text{Rep-circline-mat } (\text{opposite-ocircline-rep } H) = (-1 \ *_{sm} \ \text{Rep-circline-mat } H)$

$H)$
 $\langle \text{proof} \rangle$

lift-definition *opposite-ocircline* :: *ocircline* \Rightarrow *ocircline* **is** *opposite-ocircline-rep*
 $\langle \text{proof} \rangle$

lemma *opposite-ocircline-rep-opposite-ocircline-rep*
 $[\text{simp}]$: *opposite-ocircline-rep* (*opposite-ocircline-rep* H) = H
 $\langle \text{proof} \rangle$

lemma *opposite-ocircline-opposite-ocircline*
 $[\text{simp}]$: *opposite-ocircline* (*opposite-ocircline* H) = H
 $\langle \text{proof} \rangle$

lemma *ocircline-set-opposite-ocircline*
 $[\text{simp}]$: *ocircline-set* (*opposite-ocircline* H) = *ocircline-set* H
 $\langle \text{proof} \rangle$

lemma *disc-compl-opposite*: *disc-compl* (*opposite-ocircline* H) = *disc* H
 $\langle \text{proof} \rangle$

lemma *disc-opposite*:
disc (*opposite-ocircline* H) = *disc-compl* H
 $\langle \text{proof} \rangle$

of-ocircline, *pos-oriented*, *of-circline*

lift-definition *of-ocircline* :: *ocircline* \Rightarrow *circline* **is** *id::circline-mat* \Rightarrow *circline-mat*
 $\langle \text{proof} \rangle$

lemma *of-ocircline-opposite-ocircline* $[\text{simp}]$:
of-ocircline (*opposite-ocircline* H) = *of-ocircline* H
 $\langle \text{proof} \rangle$

lemma *circline-set-ocircline-set* $[\text{simp}]$:
circline-set (*of-ocircline* H) = *ocircline-set* H
 $\langle \text{proof} \rangle$

lemma *inj-of-ocircline*:
assumes *of-ocircline* H = *of-ocircline* H'
shows $H = H' \vee H = \text{opposite-ocircline } H'$
 $\langle \text{proof} \rangle$

lemma
inj-ocircline-set:
assumes *ocircline-set* H = *ocircline-set* H' *ocircline-set* $H \neq \{\}$
shows $H = H' \vee H = \text{opposite-ocircline } H'$
 $\langle \text{proof} \rangle$

definition *pos-oriented-rep* **where**

pos-oriented-rep $H \longleftrightarrow$
 (let $(A, B, C, D) = \text{Rep-circline-mat } H$
 in $(\text{Re } A > 0 \vee (\text{Re } A = 0 \wedge ((B \neq 0 \wedge \arg B > 0) \vee (B = 0 \wedge \text{Re } D > 0))))$)

lemma *pos-oriented-rep*: *pos-oriented-rep* $H \vee \text{pos-oriented-rep } (\text{opposite-ocircline-rep } H)$
 $\langle \text{proof} \rangle$

lift-definition *pos-oriented* :: *ocircline* \Rightarrow bool **is** *pos-oriented-rep*
 $\langle \text{proof} \rangle$

lemma *pos-oriented*: *pos-oriented* $H \vee \text{pos-oriented } (\text{opposite-ocircline } H)$
 $\langle \text{proof} \rangle$

lemma *pos-oriented-opposite-ocircline*:
pos-oriented $(\text{opposite-ocircline } H) \longleftrightarrow \neg \text{pos-oriented } H$
 $\langle \text{proof} \rangle$

lemma *pos-oriented-circle-inf*:
 assumes $\infty_h \notin \text{ocircline-set } H$
 shows *pos-oriented* $H \longleftrightarrow \infty_h \notin \text{disc } H$
 $\langle \text{proof} \rangle$

lemma
 assumes *is-circle* $(\text{of-ocircline } H) (a, r) = \text{euclidean-circle } (\text{of-ocircline } H)$
circline-type $(\text{of-ocircline } H) < 0$
 shows *pos-oriented* $H \longleftrightarrow \text{of-complex } a \in \text{disc } H$
 $\langle \text{proof} \rangle$

definition *of-circline-rep* :: *circline-mat* \Rightarrow *circline-mat* **where**
of-circline-rep $H = (\text{if } \text{pos-oriented-rep } H \text{ then } H \text{ else } \text{opposite-ocircline-rep } H)$

lift-definition *of-circline* :: *circline* \Rightarrow *ocircline* **is** *of-circline-rep*
 $\langle \text{proof} \rangle$

lemma *pos-oriented-of-circline*: *pos-oriented* $(\text{of-circline } H)$
 $\langle \text{proof} \rangle$

lemma *of-ocircline-of-circline* [simp]: *of-ocircline* $(\text{of-circline } H) = H$
 $\langle \text{proof} \rangle$

lemma *of-circline-of-ocircline-pos-oriented* [simp]:
 assumes *pos-oriented* H
 shows *of-circline* $(\text{of-ocircline } H) = H$
 $\langle \text{proof} \rangle$

lemma *ocircline-set-circline-set*[simp]: *ocircline-set* (*of-circline* *H*) = *circline-set* *H*
 ⟨*proof*⟩

lemma *inj-of-circline*:
 assumes *of-circline* *H* = *of-circline* *H'*
 shows *H* = *H'*
 ⟨*proof*⟩

lemma *of-circline-of-ocircline*:
 shows *of-circline* (*of-ocircline* *H'*) = *H' ∨ of-circline* (*of-ocircline* *H'*) = *opposite-ocircline* *H'*
 ⟨*proof*⟩

11.10 Some special oriented circlines and discs

lift-definition *mk-ocircline* :: *complex* ⇒ *complex* ⇒ *complex* ⇒ *complex* ⇒ *ocircline* **is** *mk-circline-rep*
 ⟨*proof*⟩

oriented unit circle and unit disc

lift-definition *ounit-circle* :: *ocircline* **is** *unit-circle-rep*
 ⟨*proof*⟩

definition *unit-disc* = *disc ounit-circle*

lemma *zero-in-unit-disc*: $0_h \in \text{unit-disc}$
 ⟨*proof*⟩

lemma *inf-notin-unit-disc*: $\infty_h \notin \text{unit-disc}$
 ⟨*proof*⟩

lemma *of-ocircline-ounit-circle* [simp]: *of-ocircline ounit-circle* = *unit-circle*
 ⟨*proof*⟩

lemma *of-circline-unit-circline* [simp]: *of-circline (unit-circle)* = *ounit-circle*
 ⟨*proof*⟩

Oriented x axis and lower half plane

lift-definition *o-x-axis* :: *ocircline* **is** *x-axis-rep*
 ⟨*proof*⟩

lemma *o-x-axis-pos-oriented*: *pos-oriented o-x-axis*
 ⟨*proof*⟩

lemma *of-ocircline-o-x-axis* [simp]: *of-ocircline o-x-axis* = *x-axis*
 ⟨*proof*⟩

lemma *of-circline-x-axis* [simp]: *of-circline x-axis = o-x-axis*
 ⟨proof⟩

lemma *ocircline-set-circline-set-x-axis*: *ocircline-set o-x-axis = circline-set x-axis*
 ⟨proof⟩

lemma [simp]: $i_h \notin \text{disc } o\text{-}x\text{-axis}$
 ⟨proof⟩

lemma [simp]: $i_h \in \text{disc } (\text{opposite-ocircline } o\text{-}x\text{-axis})$
 ⟨proof⟩

11.11 Moebius action on oriented circlines and discs

lift-definition *moebius-ocircline* :: *moebius* \Rightarrow *ocircline* \Rightarrow *ocircline* **is** *moebius-circline-rep*
 ⟨proof⟩

lemma *moebius-circline-ocircline*:
moebius-circline M H = of-ocircline (moebius-ocircline M (of-circline H))
 ⟨proof⟩

lemma *moebius-ocircline-circline*:
moebius-ocircline M H = of-circline (moebius-circline M (of-ocircline H)) \vee
moebius-ocircline M H = opposite-ocircline (of-circline (moebius-circline M
(of-ocircline H)))
 ⟨proof⟩

lemma
inj-moebius-ocircline: *inj (moebius-ocircline M)*
 ⟨proof⟩

lemma *moebius-ocircline-comp*:
moebius-ocircline M1 (moebius-ocircline M2 H) = moebius-ocircline (moebius-comp
M1 M2) H
 ⟨proof⟩

lemma [simp]:
moebius-ocircline id-moebius H = H
 ⟨proof⟩

lemma *moebius-ocircline-comp-inv*[simp]:
moebius-ocircline (moebius-inv M) (moebius-ocircline M H) = H
 ⟨proof⟩

lemma *moebius-circline-opposite-ocircline* [simp]:
moebius-ocircline M (opposite-ocircline H) = opposite-ocircline (moebius-ocircline
M H)
 ⟨proof⟩

lemma *moebius-ocircline-set*:

shows $\text{moebius-pt } M \text{ ' } \text{ocircline-set } H = \text{ocircline-set } (\text{moebius-ocircline } M \ H)$
(is $?lhs = ?rhs)$
 $\langle \text{proof} \rangle$

lemma *moebius-disc*:

$\text{moebius-pt } M \text{ ' } (\text{disc } H) = \text{disc } (\text{moebius-ocircline } M \ H)$
 $\langle \text{proof} \rangle$

lemma *moebius-disc-compl*:

$\text{moebius-pt } M \text{ ' } (\text{disc-compl } H) = \text{disc-compl } (\text{moebius-ocircline } M \ H)$
 $\langle \text{proof} \rangle$

lemma *similarity-preserves-lines*:

assumes $a \neq 0$
shows $\infty_h \in \text{ocircline-set } H \longleftrightarrow \infty_h \in \text{ocircline-set } (\text{moebius-ocircline } (\text{similarity-moebius } a \ b) \ H)$ **(is** $?lhs = ?rhs)$
 $\langle \text{proof} \rangle$

lemma *similarity-preserve-orientation'*:

assumes $a \neq 0 \ M = \text{similarity-moebius } a \ b \ H' = \text{moebius-ocircline } M \ H \ \infty_h \notin \text{ocircline-set } H$
shows $\text{pos-oriented } H \longrightarrow \text{pos-oriented } H'$
 $\langle \text{proof} \rangle$

lemma *similarity-preserve-orientation*:

assumes $a \neq 0 \ M = \text{similarity-moebius } a \ b \ H' = \text{moebius-ocircline } M \ H \ \infty_h \notin \text{ocircline-set } H$
shows $\text{pos-oriented } H \longleftrightarrow \text{pos-oriented } H'$
 $\langle \text{proof} \rangle$

lemma $0_h \in \text{disc-compl } (\text{mk-ocircline } -1 \ (2*ii) \ (-2*ii) \ 1)$
 $\langle \text{proof} \rangle$

lemma $\neg \text{pos-oriented } (\text{mk-ocircline } -1 \ (2*ii) \ (-2*ii) \ 1)$
 $\langle \text{proof} \rangle$

lemma $\text{circline-type } (\text{mk-circline } -1 \ (2*ii) \ (-2*ii) \ 1) = -1$
 $\langle \text{proof} \rangle$

lemma $0_h \in \text{disc-compl } (\text{mk-ocircline } 1 \ (2*ii) \ (-2*ii) \ 1)$
 $\langle \text{proof} \rangle$

lemma $\text{pos-oriented } (\text{mk-ocircline } 1 \ (2*ii) \ (-2*ii) \ 1)$
 $\langle \text{proof} \rangle$

lemma $\text{circline-type } (\text{mk-circline } 1 \ (2*ii) \ (-2*ii) \ 1) = -1$
 $\langle \text{proof} \rangle$

lemma *reciprocal-preserve-orientation*:

assumes $0_h \in \text{disc-compl } H \ M = \text{reciprocal-moebius } H' = \text{moebius-ocircline } M$

H
shows *pos-oriented* H'
 $\langle \text{proof} \rangle$

lemma *reciprocal-not-preserve-orientation*:
assumes $0_h \in \text{disc } H$ $M = \text{reciprocal-moebius } H' = \text{moebius-ocircline } M H$
shows $\neg \text{pos-oriented } H'$
 $\langle \text{proof} \rangle$

lemma *pole-in-disc*:
assumes $M = \text{mk-moebius } a b c d$ $c \neq 0$ $a*d - b*c \neq 0$
assumes *is-pole* $M z z \in \text{disc } H$ $H' = \text{moebius-ocircline } M H$
shows $\neg \text{pos-oriented } H'$
 $\langle \text{proof} \rangle$

lemma *pole-in-disc-compl*:
assumes $M = \text{mk-moebius } a b c d$ $c \neq 0$ $a*d - b*c \neq 0$
assumes *is-pole* $M z z \in \text{disc-compl } H$ $H' = \text{moebius-ocircline } M H$
shows *pos-oriented* H'
 $\langle \text{proof} \rangle$

11.12 Oriented circlines uniqueness

lemma *ocircline-01inf*:
assumes $0_h \in \text{ocircline-set } H \wedge 1_h \in \text{ocircline-set } H \wedge \infty_h \in \text{ocircline-set } H$
shows $H = \text{o-x-axis} \vee H = \text{opposite-ocircline o-x-axis}$
 $\langle \text{proof} \rangle$

lemma *unique-ocircline-01inf*: $\exists! H. 0_h \in \text{ocircline-set } H \wedge 1_h \in \text{ocircline-set } H$
 $\wedge \infty_h \in \text{ocircline-set } H \wedge i_h \notin \text{disc } H$
 $\langle \text{proof} \rangle$

lemma *unique-ocircline-set*:
assumes $A \neq B$ $A \neq C$ $B \neq C$
shows $\exists! H. \text{pos-oriented } H \wedge (A \in \text{ocircline-set } H \wedge B \in \text{ocircline-set } H \wedge C \in \text{ocircline-set } H)$
 $\langle \text{proof} \rangle$

definition *chordal-circle-rep* where

chordal-circle-rep $a r =$
 $(\text{let } (a1, a2) = \text{Rep-homo-coords } a \text{ in}$
 $\text{mk-circline-rep } (4*a2*cnj a2 - (\text{cor } r)^2*(a1*cnj a1 + a2*cnj a2)) (-4*a1*cnj$
 $a2) (-4*cnj a1*a2) (4*a1*cnj a1 - (\text{cor } r)^2*(a1*cnj a1 + a2*cnj a2)))$

lemma *[simp]*: $\text{Rep-circline-mat } (\text{chordal-circle-rep } a r) = (\text{let } (a1, a2) = \text{Rep-homo-coords } a \text{ in}$
 $(4*a2*cnj a2 - (\text{cor } r)^2*(a1*cnj a1 + a2*cnj a2), -4*a1*cnj a2, -4*cnj$
 $a1*a2, 4*a1*cnj a1 - (\text{cor } r)^2*(a1*cnj a1 + a2*cnj a2)))$

$\langle proof \rangle$

lift-definition *chordal-circle* :: *complex-homo* \Rightarrow *real* \Rightarrow *circline* **is** *chordal-circle-rep*
 $\langle proof \rangle$

lemma *sqrt-1-plus-square*: $\text{sqrt } (1 + a^2) \neq 0$
 $\langle proof \rangle$

lemma
 assumes *dist-homo* $z \ a = r$
 shows $z \in \text{circline-set } (\text{chordal-circle } a \ r)$
 $\langle proof \rangle$

lemma [*simp*]: $\text{sqrt } 4 = 2$
 $\langle proof \rangle$

lemma
 assumes $z \in \text{circline-set } (\text{chordal-circle } a \ r) \ r \geq 0$
 shows *dist-homo* $z \ a = r$
 $\langle proof \rangle$

lemma *chordal-circle-radius-positive*:
 assumes *hermitean* $(A, B, C, D) \ \text{Re } (\text{mat-det } (A, B, C, D)) \leq 0 \ B \neq 0$
 $\text{dsc} = \text{sqrt}(\text{Re } ((D-A)^2 + 4 * (B * \text{cnj } B))) \ a1 = (A - D + \text{cor dsc}) / (2 * C)$
 $a2 = (A - D - \text{cor dsc}) / (2 * C)$
 shows $\text{Re } (A * a1 / B) \geq -1 \wedge \text{Re } (A * a2 / B) \geq -1$
 $\langle proof \rangle$

definition *chordal-circles-rep* **where**
 chordal-circles-rep $H =$
 $(\text{let } (A, B, C, D) = \text{Rep-circline-mat } H;$
 $\text{dsc} = \text{sqrt}(\text{Re } ((D-A)^2 + 4 * (B * \text{cnj } B)));$
 $a1 = (A - D + \text{cor dsc}) / (2 * C);$
 $r1 = \text{sqrt}((4 - \text{Re}((-4 * a1 / B) * A)) / (1 + \text{Re } (a1 * \text{cnj } a1)));$
 $a2 = (A - D - \text{cor dsc}) / (2 * C);$
 $r2 = \text{sqrt}((4 - \text{Re}((-4 * a2 / B) * A)) / (1 + \text{Re } (a2 * \text{cnj } a2)));$
 $\text{in } ((a1, r1), (a2, r2)))$

lift-definition *chordal-circles* :: *ocircline* \Rightarrow $(\text{complex} \times \text{real}) \times (\text{complex} \times \text{real})$
is *chordal-circles-rep*
 $\langle proof \rangle$

lemma *chordal-circle-det-positive*:
 fixes $x \ y :: \text{real}$
 assumes $x * y < 0$
 shows $x / (x - y) > 0$
 $\langle proof \rangle$

lemma *chordal-circle1*:

assumes *is-real A is-real D* $\text{Re } (A * D) < 0$ $r = \text{sqrt}(\text{Re } ((4 * A) / (A - D)))$
shows *mk-circline A 0 0 D = chordal-circle* $\infty_h r$
 <proof>

lemma *chordal-circle2*:
assumes *is-real A is-real D* $\text{Re } (A * D) < 0$ $r = \text{sqrt}(\text{Re } ((4 * D) / (D - A)))$
shows *mk-circline A 0 0 D = chordal-circle* $0_h r$
 <proof>

lemma *chordal-circle'*:
assumes $B \neq 0$ $(A, B, C, D) \in \{H. \text{hermitean } H \wedge H \neq \text{mat-zero}\}$ $\text{Re } (\text{mat-det } (A, B, C, D)) \leq 0$
 $C * a^2 + (D - A) * a - B = 0$ $r = \text{sqrt}((4 - \text{Re}((-4 * a / B) * A)) / (1 + \text{Re } (a * \text{cnj } a)))$
shows *mk-circline A B C D = chordal-circle (of-complex a)* r
 <proof>

lift-definition *o-circline-point-0h* :: *ocircline* **is** *circline-point-0h-rep*
 <proof>

lemma *of-ocircline-o-circline-point-0h [simp]*: *of-ocircline o-circline-point-0h = circline-point-0h*
 <proof>

lemma *ocircline-set-0h*:
assumes *ocircline-set H = {0h}*
shows $H = \text{o-circline-point-0h} \vee H = \text{opposite-ocircline } (\text{o-circline-point-0h})$
 <proof>

11.13 Disc automorphisms

lemma *circline-set-fix-iff-circline-fix*:
assumes *circline-set H' ≠ {}*
shows $(\text{moebius-pt } M) \text{ ' } (\text{circline-set } H) = \text{circline-set } H' \longleftrightarrow \text{moebius-circline } M H = H'$
 <proof>

lemma *ocircline-set-fix-iff-ocircline-fix*:
assumes *ocircline-set H' ≠ {}*
shows $(\text{moebius-pt } M) \text{ ' } (\text{ocircline-set } H) = \text{ocircline-set } H' \longleftrightarrow$
 $\text{moebius-ocircline } M H = H' \vee \text{moebius-ocircline } M H = \text{opposite-ocircline } H'$
 <proof>

definition *Unitary11-gen-rep* **where**
 $\text{Unitary11-gen-rep } M \longleftrightarrow \text{unitary11-gen } (\text{Rep-moebius-mat } M)$

lift-definition *Unitary11-gen* :: *moebius* \Rightarrow *bool* **is** *Unitary11-gen-rep*
 <proof>

lemma *unit-circle-fix-iff-Unitary11-gen*:
shows *moebius-circline* *M* *unit-circle* = *unit-circle* \longleftrightarrow *Unitary11-gen* *M* (**is** ?lhs
= ?rhs)
 \langle proof \rangle

lemma *unit-circle-set-fix-iff-Unitary11-gen*:
shows (*moebius-pt* *M* ' (*circline-set* *unit-circle*) = (*circline-set* *unit-circle*)) \longleftrightarrow
Unitary11-gen *M* (**is** ?lhs \longleftrightarrow ?rhs)
 \langle proof \rangle

definition *Unitary11-gen-direct-rep* **where**
Unitary11-gen-direct-rep *M* \longleftrightarrow
(let (*A*, *B*, *C*, *D*) = *Rep-moebius-mat* *M*
in *unitary11-gen* (*A*, *B*, *C*, *D*) \wedge (*B* = 0 \vee *Re* ((*A***D*)/(*B***C*)) > 1))

lift-definition *Unitary11-gen-direct* :: *moebius* \Rightarrow *bool* **is** *Unitary11-gen-direct-rep*
 \langle proof \rangle

lemma *ounit-circle-fix-iff-Unitary11-gen-direct*:
shows *moebius-ocircline* *M* *ounit-circle* = *ounit-circle* \longleftrightarrow *Unitary11-gen-direct*
M (**is** ?lhs \longleftrightarrow ?rhs)
 \langle proof \rangle

Blaschke factor

definition *blaschke-rep* **where**
blaschke-rep *a* = *Abs-moebius-mat* (1, -*a*, -*cnj* *a*, 1)

lemma *blaschke-rep-Rep1*:
assumes *cmod* *a* \neq 1
shows *Rep-moebius-mat* (*blaschke-rep* *a*) = (1, -*a*, -*cnj* *a*, 1)
 \langle proof \rangle

lemma *blaschke-rep-Rep2*:
assumes *a* * *cnj* *a* \neq 1
shows *Rep-moebius-mat* (*blaschke-rep* *a*) = (1, -*a*, -*cnj* *a*, 1)
 \langle proof \rangle

lift-definition *blaschke* :: *complex* \Rightarrow *moebius* **is** *blaschke-rep*
 \langle proof \rangle

lemma *blaschke-a-to-zero*:
assumes *cmod* *a* \neq 1
shows *moebius-pt* (*blaschke* *a*) (*of-complex* *a*) = 0_{*h*}
 \langle proof \rangle

lemma *blaschke-inv-a-inf*:
assumes *cmod* *a* \neq 1
shows *moebius-pt* (*blaschke* *a*) (*inversion-homo* (*of-complex* *a*)) = ∞_h

<proof>

lemma *blaschke-Unitary11-gen-rep*:
 assumes $a * \text{cnj } a \neq 1$
 shows *Unitary11-gen-rep* (*blaschke-rep a*)
<proof>

lemma *blaschke-unitary11-gen-direct-rep*:
 assumes $\text{Re } (a * \text{cnj } a) < 1$
 shows *Unitary11-gen-direct-rep* (*blaschke-rep a*)
<proof>

lemma *blaschke-Unitary11-gen*:
 assumes $a * \text{cnj } a \neq 1$
 shows *Unitary11-gen* (*blaschke a*)
<proof>

lemma *blaschke-Unitary11-gen-direct*:
 assumes $\text{Re } (a * \text{cnj } a) < 1$
 shows *Unitary11-gen-direct* (*blaschke a*)
<proof>

lemma *blaschke-unit-circle-fix*:
 assumes $\text{cmod } a \neq 1$
 shows *moebius-circline* (*blaschke a*) *unit-circle* = *unit-circle*
<proof>

lemma *blaschke-ounit-circle-fix*:
 assumes $\text{cmod } a < 1$
 shows *moebius-ocircline* (*blaschke a*) *ounit-circle* = *ounit-circle*
<proof>

lemma [*simp*]: *hermitean* $(1, 0, 0, -1)$
<proof>

definition *is-disc-aut* **where** *is-disc-aut* $f \longleftrightarrow \text{bij-betw } f \text{ unit-disc unit-disc}$

lemma *is-disc-aut-iff-unit-disc-fix*:
 shows *is-disc-aut* (*moebius-pt M*) $\longleftrightarrow (\text{moebius-pt } M) ' \text{unit-disc} = \text{unit-disc}$
<proof>

lemma *comp-inv-l*:
 assumes $f \circ \text{inv } g = h \text{ bij } g$
 shows $f = h \circ g$
<proof>

lemma *in-unit-disc-cmod-lt-1*:
 assumes *of-complex* $a \in \text{unit-disc}$

shows $cmod\ a < 1$
 $\langle proof \rangle$

11.14 Angle between circlines

fun *mat-det-12* :: *complex-mat* \Rightarrow *complex-mat* \Rightarrow *complex* **where**
 $mat-det-12\ (A1,\ B1,\ C1,\ D1)\ (A2,\ B2,\ C2,\ D2) = A1*D2 + A2*D1 - B1*C2 - B2*C1$

lemma *mat-det-12-mm-l* [*simp*]: $mat-det-12\ (M\ *_{mm}\ A)\ (M\ *_{mm}\ B) = mat-det\ M\ *\ mat-det-12\ A\ B$
 $\langle proof \rangle$

lemma *mat-det-12-mm-r* [*simp*]: $mat-det-12\ (A\ *_{mm}\ M)\ (B\ *_{mm}\ M) = mat-det\ M\ *\ mat-det-12\ A\ B$
 $\langle proof \rangle$

lemma *mat-det-12-sm-l* [*simp*]: $mat-det-12\ (k\ *_{sm}\ A)\ B = k\ *\ mat-det-12\ A\ B$
 $\langle proof \rangle$

lemma *mat-det-12-sm-r* [*simp*]: $mat-det-12\ A\ (k\ *_{sm}\ B) = k\ *\ mat-det-12\ A\ B$
 $\langle proof \rangle$

lemma *mat-det-12-congruence* [*simp*]:
 $mat-det-12\ (congruence\ M\ A)\ (congruence\ M\ B) = (cor\ ((cmod\ (mat-det\ M))^2))\ *\ mat-det-12\ A\ B$
 $\langle proof \rangle$

lemma *mat-det-congruence* [*simp*]:
 $mat-det\ (congruence\ M\ A) = (cor\ ((cmod\ (mat-det\ M))^2))\ *\ mat-det\ A$
 $\langle proof \rangle$

definition *cos-angle-rep* **where**
 $cos-angle-rep\ H1\ H2 =$
 $(let\ H1 = Rep-circline-mat\ H1;$
 $\quad H2 = Rep-circline-mat\ H2\ in$
 $\quad -\ Re\ (mat-det-12\ H1\ H2)\ /\ (2\ * (sqrt\ (Re\ (mat-det\ H1\ *\ mat-det\ H2))))))$

lemma [*simp*]: $is-real\ (mat-det\ (Rep-circline-mat\ H))$
 $\langle proof \rangle$

lift-definition *cos-angle* :: *ocircline* \Rightarrow *ocircline* \Rightarrow *real* **is** *cos-angle-rep*
 $\langle proof \rangle$

lemma *ang-vec-opposite-opposite'*:
assumes $a1 \neq E\ a2 \neq E$
shows $(E - a1)\ (E - a2) = (a1 - E)\ (a2 - E)$
 $\langle proof \rangle$

lemma *cos-ang-circ-simp*:
assumes $E \neq \mu 1$ $E \neq \mu 2$
shows $\cos (\text{ang-circ } E \ \mu 1 \ \mu 2 \ p1 \ p2) = \text{sgn-bool } (p1 = p2) * \cos (\arg (E - \mu 2) - \arg (E - \mu 1))$
 $\langle \text{proof} \rangle$

lemma *Re-sgn*:
assumes *is-real* A $A \neq 0$
shows $\text{Re } (\text{sgn } A) = \text{sgn-bool } (\text{Re } A > 0)$
 $\langle \text{proof} \rangle$

lemma *Re-mult-real3*:
assumes *is-real* $z1$ *is-real* $z2$ *is-real* $z3$
shows $\text{Re } (z1 * z2 * z3) = \text{Re } z1 * \text{Re } z2 * \text{Re } z3$
 $\langle \text{proof} \rangle$

lemma [*simp*]: $\text{sgn } (\text{sqrt } x) = \text{sgn } x$
 $\langle \text{proof} \rangle$

lemma *sgn-divide*:
fixes $x \ y :: \text{real}$
shows $\text{sgn } (x / y) = \text{sgn } x / \text{sgn } y$
 $\langle \text{proof} \rangle$

lemma *real-circle-sgn-r*:
assumes *is-circle* H $(a, r) = \text{euclidean-circle } H$
shows $\text{sgn } r = - \text{circline-type } H$
 $\langle \text{proof} \rangle$

lemma
assumes
is-circle (*of-ocircline* $H1$) *is-circle* (*of-ocircline* $H2$)
circline-type (*of-ocircline* $H1$) < 0 *circline-type* (*of-ocircline* $H2$) < 0
 $(a1, r1) = \text{euclidean-circle } (\text{of-ocircline } H1)$ $(a2, r2) = \text{euclidean-circle } (\text{of-ocircline } H2)$
of-complex $E \in \text{ocircline-set } H1 \cap \text{ocircline-set } H2$
shows $\cos\text{-angle } H1 \ H2 = \cos (\text{ang-circ } E \ a1 \ a2 \ (\text{pos-oriented } H1) \ (\text{pos-oriented } H2))$
 $\langle \text{proof} \rangle$

lemma [*simp*]: $\text{sqrt } a * \text{sqrt } a = |a|$
 $\langle \text{proof} \rangle$

lemma $\cos\text{-angle } H1 \ H2 = \cos\text{-angle } (\text{moebius-ocircline } M \ H1) \ (\text{moebius-ocircline } M \ H2)$
 $\langle \text{proof} \rangle$


```

lemma
  assumes mat-det (A, B, C, D)  $\neq 0$ 
  shows moebius-circline (mk-moebius A B C D) imag-unit-circle = imag-unit-circle
 $\longleftrightarrow$ 
    unitary-gen (A, B, C, D) (is ?lhs = ?rhs)
 $\langle proof \rangle$ 

end

```