Morley's Theorem*

Benjamin Puyobro

Université Paris Saclay, IRT SystemX, LMF, CNRS

April 2, 2025

Contents

*This work has been supported by the French government under the "France 2030" program, as part of the SystemX Technological Research Institute within the CVH project.

theory *Complex-Angles*

imports *Complex-Geometry.Elementary-Complex-Geometry*

begin

1 Introduction

Morley's trisector theorem states that in any triangle, the three points of intersection of the adjacent angle trisectors form an equilateral triangle, called the first Morley triangle or simply the Morley triangle. This theorem is listed in the 100 theorem list [3].

In this theory, we define some basics elements on complex geometry such as axial symmetry, rotations. We also define basic property of congruent triangles in the complex field following the model already presented in the afp [1] In addition we demonstrate sines law in the complex context.

Finally we prove the Morley theorem using Alain Connes's proof [2].

2 Angles between three complex

definition *angle-c-def*:‹*angle-c a b c* ≡ ∠ (*a*−*b*) (*c*−*b*)›

lemma *angle-c-commute-pi*:
  assumes ‹*angle-c a b c* = *pi*›
  shows *angle-c a b c* = *angle-c c b a*
  ⟨*proof*⟩

lemma *angle-c-commute*:
  assumes ‹*angle-c a b c* ≠ *pi*›
  shows *angle-c a b c* = −*angle-c c b a*
  ⟨*proof*⟩

lemma *angle-c-sum*:
  assumes *1*:‹*a* − *b*≠*0*› **and** *2*:‹*c* − *b*≠ *0*› **and** *3*:‹*b*−*d* ≠ *0*›
  shows‹⌊*angle-c a b c* + *angle-c c b d*⌋ = *angle-c a b d*›
  ⟨*proof*⟩

lemma *collinear-angle*:‹*collinear a b c* ⟹ *a*≠*b* ⟹ *b*≠*c* ⟹ *c*≠*a* ⟹ *angle-c a b c* = *0* ∨ *angle-c a b c* = *pi*›
  ⟨*proof*⟩

lemma *cdist-commute*:‹*cdist a b* = *cdist b a*›
  ⟨*proof*⟩

lemma *angle-sum-triangle*:

assumes $h{:}a \neq b \wedge b \neq c \wedge a \neq c$
shows $\lfloor\angle\ (c-a)\ (b-a) + \angle\ (a-b)\ (c-b)\ + \angle\ (b-c)\ (a-c)\rfloor = pi$
⟨*proof*⟩

lemma *angle-sum-triangle-c*:
assumes $h{:}a \neq b \wedge b \neq c \wedge a \neq c$
shows $\lfloor angle\text{-}c\ c\ a\ b + angle\text{-}c\ a\ b\ c\ + angle\text{-}c\ b\ c\ a\rfloor = pi$
⟨*proof*⟩

lemma *angle-sum-triangle′*:
assumes $h{:}a \neq 0 \wedge b \neq 0 \wedge c \neq 0$
shows $\lfloor\angle\ (-a)\ b + \angle\ (-b)\ c\ + \angle\ (-c)\ a\rfloor = pi$
⟨*proof*⟩

lemma *ang-c-in*:‹$angle\text{-}c\ a\ b\ c \in \{-pi{<}..pi\}$›
⟨*proof*⟩

lemma *angle-c-aac*:‹$angle\text{-}c\ a\ a\ c = \lfloor Arg\ (c-a)\rfloor$›
⟨*proof*⟩

lemma *angle-c-caa*:‹$angle\text{-}c\ c\ a\ a = \lfloor -Arg\ (c-a)\rfloor$›
⟨*proof*⟩

lemma *angle-c-neq0*:‹$angle\text{-}c\ p\ q\ r \neq 0 \implies p{\neq}r$ ›
⟨*proof*⟩

end
theory *Complex-Triangles-Definitions*

imports *Complex-Angles*

begin

3 Complex triangles

In this section we define triangles and derive some useful lemmas on congruent triangles, following the model [1]

locale *congruent-ctriangle* =
fixes *a1 b1 c1* :: *complex* **and** *a2 b2 c2* :: *complex*
assumes *sides′*: $cdist\ a1\ b1 = cdist\ a2\ b2\ cdist\ a1\ c1 = cdist\ a2\ c2\ cdist\ b1\ c1 = cdist\ b2\ c2$
and *angles′*: $angle\text{-}c\ b1\ a1\ c1 = angle\text{-}c\ b2\ a2\ c2 \vee angle\text{-}c\ b1\ a1\ c1 = -\ angle\text{-}c\ b2\ a2\ c2$
$angle\text{-}c\ a1\ b1\ c1 = angle\text{-}c\ a2\ b2\ c2 \vee angle\text{-}c\ a1\ b1\ c1 = -\ angle\text{-}c\ a2\ b2\ c2$
$angle\text{-}c\ a1\ c1\ b1 = angle\text{-}c\ a2\ c2\ b2 \vee angle\text{-}c\ a1\ c1\ b1 = -\ angle\text{-}c\ a2\ c2\ b2$
begin

lemma *sides*:
cdist a1 b1 = *cdist a2 b2 cdist a1 c1* = *cdist a2 c2 cdist b1 c1* = *cdist b2 c2*
cdist b1 a1 = *cdist a2 b2 cdist c1 a1* = *cdist a2 c2 cdist c1 b1* = *cdist b2 c2*
cdist a1 b1 = *cdist b2 a2 cdist a1 c1* = *cdist c2 a2 cdist b1 c1* = *cdist c2 b2*
cdist b1 a1 = *cdist b2 a2 cdist c1 a1* = *cdist c2 a2 cdist c1 b1* = *cdist c2 b2*
⟨*proof*⟩

lemma *angles*:
angle-c b1 a1 c1 = *angle-c b2 a2 c2* ∨ *angle-c b1 a1 c1* = − *angle-c b2 a2 c2*
angle-c a1 b1 c1 = *angle-c a2 b2 c2* ∨ *angle-c a1 b1 c1* = − *angle-c a2 b2 c2*
angle-c a1 c1 b1 = *angle-c a2 c2 b2* ∨ *angle-c a1 c1 b1* = − *angle-c a2 c2 b2*
angle-c c1 a1 b1 = *angle-c b2 a2 c2* ∨ *angle-c c1 a1 b1* = − *angle-c b2 a2 c2*
angle-c c1 b1 a1 = *angle-c a2 b2 c2* ∨ *angle-c c1 b1 a1* = − *angle-c a2 b2 c2*
angle-c b1 c1 a1 = *angle-c a2 c2 b2* ∨ *angle-c b1 c1 a1* = − *angle-c a2 c2 b2*
angle-c b1 a1 c1 = *angle-c c2 a2 b2* ∨ *angle-c b1 a1 c1* = − *angle-c c2 a2 b2*
angle-c a1 b1 c1 = *angle-c c2 b2 a2* ∨ *angle-c a1 b1 c1* = − *angle-c c2 b2 a2*
angle-c a1 c1 b1 = *angle-c b2 c2 a2* ∨ *angle-c a1 c1 b1* = − *angle-c b2 c2 a2*
angle-c c1 a1 b1 = *angle-c c2 a2 b2* ∨ *angle-c c1 a1 b1* = − *angle-c c2 a2 b2*
angle-c c1 b1 a1 = *angle-c c2 b2 a2* ∨ *angle-c c1 b1 a1* = − *angle-c c2 b2 a2*
angle-c b1 c1 a1 = *angle-c b2 c2 a2* ∨ *angle-c b1 c1 a1* = − *angle-c b2 c2 a2*
⟨*proof*⟩

end

end
theory *Complex-Trigonometry*

imports *HOL−Analysis.Convex Complex-Angles Complex-Triangles-Definitions*

begin

4 Complex trigonometry

In this section we add some trigonometric results, especially the law of sines

lemma *ang-sin*:
shows‹*Im* ((*b*−*a*)∗*cnj*(*c*−*a*)) = *cmod* (*c*−*a*) ∗*cmod* (*b*−*a*) ∗ *sin* (*angle-c c a b*)›
⟨*proof*⟩

lemma *ang-cos*:
shows‹*Re* ((*b*−*a*)∗*cnj*(*c*−*a*)) = *cmod* (*c*−*a*) ∗*cmod* (*b*−*a*) ∗ *cos* (*angle-c c a b*)›
⟨*proof*⟩

lemma *law-of-cosines′*:
assumes *h*:‹*A*≠*C*› ‹*A*≠*B*›
shows $((cdist\ B\ C)^2 - (cdist\ A\ C)^2 - (cdist\ A\ B)^2\ ) / (-\ 2*(cdist\ A\ C)*(cdist\ A\ B)) = (cos\ (\angle\ (C-A)\ (B-A)))$

⟨*proof*⟩

lemma *law-of-cosines''*:
shows $(cdist\ A\ C)^2 = (cdist\ B\ C)^2 - (cdist\ A\ B)^2 + 2*(cdist\ A\ C)*(cdist\ A\ B)*(cos\ (\angle\ (C-A)\ (B-A)))$
⟨*proof*⟩

lemma *law-of-cosines'''*:
shows $(cdist\ A\ B)^2 = (cdist\ B\ C)^2 - (cdist\ A\ C)^2 + 2*(cdist\ A\ C)*(cdist\ A\ B)*(cos\ (\angle\ (C-A)\ (B-A)))$
⟨*proof*⟩

4.1 The law of sines

theorem *law-of-sines*:
assumes *h1*:‹$b \neq a$› ‹$a \neq c$› ‹$b \neq c$›
shows $sin\ (angle\text{-}c\ a\ b\ c) * cdist\ b\ c = sin\ (angle\text{-}c\ c\ a\ b) * cdist\ a\ c$ (**is** *?A* = *?B*)
⟨*proof*⟩

lemma *law-of-sines'*: **assumes** *h1*:‹$b \neq a$› ‹$a \neq c$› ‹$b \neq c$›
shows $sin\ (angle\text{-}c\ a\ b\ c) * cdist\ b\ a = sin\ (angle\text{-}c\ b\ c\ a) * cdist\ a\ c$
⟨*proof*⟩

lemma *ang-pos-pos*:‹$q\neq p \Longrightarrow p\neq r \Longrightarrow r \neq q \Longrightarrow angle\text{-}c\ q\ r\ p \geq 0 \Longrightarrow angle\text{-}c\ r\ p\ q \geq 0$›
⟨*proof*⟩

lemma *cmod-pos*:‹$cmod\ a \geq 0$›
⟨*proof*⟩

lemma *ang-neg-neg*:‹$q\neq p \Longrightarrow p\neq r \Longrightarrow r \neq q \Longrightarrow angle\text{-}c\ q\ r\ p < 0 \Longrightarrow angle\text{-}c\ r\ p\ q < 0$›
⟨*proof*⟩

lemma *collinear-sin-neq-0*:
‹$\neg collinear\ a2\ b2\ c2 \Longrightarrow sin\ (angle\text{-}c\ a2\ c2\ b2) \neq 0$›
⟨*proof*⟩

lemma *collinear-sin-neq-pi*:
‹$\neg collinear\ a2\ b2\ c2 \Longrightarrow sin\ (angle\text{-}c\ a2\ c2\ b2) \neq pi$›
⟨*proof*⟩

lemma *collinear-iff*:
assumes ‹$a\neq b \wedge b\neq c \wedge c\neq a$›
shows ‹$collinear\ a\ b\ c \longleftrightarrow (angle\text{-}c\ a\ b\ c = pi \vee angle\text{-}c\ a\ b\ c = 0)$›

⟨proof⟩

definition ‹*innerprod a b* ≡ *cnj a* ∗ *b*›

lemma *left-lin-innerprod*:‹*innerprod* (*x* + *y*) *z* = *innerprod x z* + *innerprod y z*›
⟨proof⟩

lemma *right-lin-innerprod*:‹*innerprod x* (*y*+*z*) = *innerprod x y* + *innerprod x z*›
⟨proof⟩

lemma *leftlin-innerprod*:‹*innerprod x* (*t*∗*y*) = *t* ∗ *innerprod x y*›
⟨proof⟩

lemma *rightsesqlin-innerprod*:‹*innerprod* (*t*∗*x*) (*y*) = *cnj t* ∗ *innerprod x y*›
⟨proof⟩

lemma *norm-eq-csqrt-inner*:‹*norm x* = *csqrt* (*innerprod x x*)›
⟨proof⟩

lemma *abs2-eq-inner*:‹*abs* (*innerprod x y*)^*2* = *innerprod x y* ∗ *cnj* (*innerprod x y*)›
⟨proof⟩

lemma *complex-add-inner-cnj*:‹*t*∗*innerprod x y* + *cnj* (*t*∗*innerprod x y*) = *2*∗ *Re* (*t*∗*innerprod x y*)›
⟨proof⟩

lemma *Re-innerprod-inner*:‹*Re* (*innerprod* (*a*−*b*) (*c*−*b*)) = (*a*−*b*)•(*c*−*b*)›
⟨proof⟩

lemma *angle-c-arccos-pos*:
assumes *h*:‹*a*≠*b* ∧ *b*≠ *c*› ‹*angle-c a b c* ≥ *0*›
shows ‹*angle-c a b c* = *arccos* ((*Re* (*innerprod* (*a*−*b*) (*c* − *b*)))/(*cmod*(*a*−*b*)∗*cmod*(*c*−*b*)))›
⟨proof⟩

lemma *angle-c-arccos-neg*:
assumes *h*:‹*a*≠*b* ∧ *b*≠ *c*› ‹*angle-c a b c* ≤ *0* ›
shows ‹− *angle-c a b c* = *arccos* ((*Re* (*innerprod* (*a*−*b*) (*c* − *b*)))/(*cmod*(*a*−*b*)∗*cmod*(*c*−*b*)))›
⟨proof⟩

end
theory *Third-Unity-Root*

imports *Complex-Angles*

begin

5 Third unity root

In this section we prove some basic properties of the third unity root j.

lemma *root-unity-3*: ‹(*z*::*complex*)^*3* − *1* = *0* ⟷ (*z* = *cis* (*2*∗*pi*/*3*) ∨ *z*=*1* ∨ *z* = *cis* (*4*∗*pi*/*3*))›
⟨*proof*⟩

definition *root3* **where** ‹*root3*≡*cis*(*2*∗*pi*/*3*)›

lemma *root3-eq-0*:‹*1* + *root3* + *root3*^*2* = *0*›
⟨*proof*⟩

lemma *root-unity-carac*:
 assumes *h1*:‹*a1*∗*a2*∗*a3* = *j*› ‹*1*−*a1*∗*a2* ≠ *0* ∧ *1*−*a2*∗*a3*≠*0* ∧ *1*−*a1*∗*a3* ≠ *0*›
‹*j*=*root3*›
  and *R* : ‹*R*=(*a1* ∗ *b2* + *b1*) / (*1* − *a1* ∗ *a2*)› (**is** ‹*R*=*?R*›)
  and *P* :‹*P*=(*a2*∗*b3* + *b2*)/(*1*−*a2*∗*a3*)› (**is** ‹*P*=*?P*›)
  and *Q*:‹*Q*=(*a3*∗*b1* + *b3*)/(*1*−*a3*∗*a1*)› (**is** ‹*Q*=*?Q*›)
 shows ‹(*a1*^*2*+*a1*+*1*)∗*b1* + *a1*^*3*∗(*a2*^*2*+*a2*+*1*)∗*b2* +*a1*^*3*∗*a2*^*3*∗(*a3*^*2*+*a3*+*1*)∗*b3* =
   −*j*∗*a1*^*2*∗*a2*∗(*a1*−*j*)∗(*a2*−*j*)∗(*a3*−*j*)∗(*?R* +*j*∗*?P* +*j*^*2*∗*?Q*)›
⟨*proof*⟩

end
theory *Complex-Triangles*

imports *Complex-Trigonometry Third-Unity-Root*

begin

lemma *similar-triangles′*:
 assumes *h*:*a* ≠ *0* ∧ *b* ≠ *0* ∧ *0* ≠ *c* ∧ *a′* ≠ *0* ∧ *b′*≠*0* ∧ *c′*≠*0*
  and *h1*:‹∠ (−*a*) *b* = ∠ (−*a′*) *b′*› ‹∠ (−*b′*) *c′* = ∠ (−*b*) *c*›
 shows ‹∠ (−*c*) *a* = ∠ (−*c′*) *a′*›
⟨*proof*⟩

lemma *similar-triangles*:
 assumes *h*:*a* ≠ *b* ∧ *b* ≠ *c* ∧ *a* ≠ *c* ∧ *a′* ≠ *b′* ∧ *b′*≠ *c′* ∧ *c′*≠*a′*
  and *h1*:‹∠ (*c*−*a*) (*b*−*a*) = ∠ (*c′*−*a′*) (*b′*−*a′*)› ‹∠ (*a*−*b*) (*c*−*b*) = ∠ (*a′*−*b′*) (*c′*−*b′*)›
 shows ‹ ∠ (*b*−*c*) (*a*−*c*) = ∠ (*b′*−*c′*) (*a′*−*c′*)›
⟨*proof*⟩

lemma *similar-triangles-c*:

assumes *h*:$a \neq b \wedge b \neq c \wedge a \neq c \wedge a' \neq b' \wedge b' \neq c' \wedge c' \neq a'$
and *h1*:‹*angle-c c a b* = *angle-c c' a' b'*› ‹*angle-c a b c* = *angle-c a' b' c'*›
shows ‹*angle-c b c a* = *angle-c b' c' a'*›
⟨*proof*⟩

lemmas *congruent-ctriangleD* = *congruent-ctriangle.sides congruent-ctriangle.angles*

lemma *congruent-ctriangles-sss*:
assumes *h*:$a \neq b \wedge b \neq c \wedge a \neq c$
and *h1*:‹*cmod* $(b - a)$= *cmod* $(b' - a')$› ‹*cmod* $(b - c)$ = *cmod* $(b' - c')$› ‹*cmod* $(c - a)$ = *cmod* $(c' - a')$›
shows ‹*congruent-ctriangle a b c a' b' c'*›
⟨*proof*⟩

lemma *congruent-ctriangleI-sss*:
assumes *h*:$a \neq b \wedge b \neq c \wedge a \neq c$
and *h1*:‹*cdist a b* = *cdist a' b'*› ‹*cdist b c* = *cdist b' c'*› ‹*cdist a c* = *cdist a' c'*›
shows ‹*congruent-ctriangle a b c a' b' c'*›
⟨*proof*⟩

lemmas *congruent-ctriangle-sss* = *congruent-ctriangleD*[*OF congruent-ctriangleI-sss*]

lemma *isosceles-triangles*:
assumes ‹*cdist a b* = *cdist b c*›
shows ‹*angle-c b c a* = *angle-c b a c* ∨ *angle-c b c a* = − *angle-c b a c* ›
⟨*proof*⟩

lemma *non-collinear-independant*:¬ *collinear a b c* ⟹ $a \neq b \wedge b \neq c \wedge a \neq c$
⟨*proof*⟩

lemma *congruent-ctriangleI-sas*:
assumes ‹*a1* ≠ *b1* ∧ *b1* ≠ *c1* ∧ *a1* ≠ *c1*›
assumes *h1*:*cdist a1 b1* = *cdist a2 b2*
assumes *h2*:*cdist b1 c1* = *cdist b2 c2*
assumes *h3*:*angle-c a1 b1 c1* = *angle-c a2 b2 c2* ∨ *angle-c a1 b1 c1* = − *angle-c a2 b2 c2*
shows *congruent-ctriangle a1 b1 c1 a2 b2 c2*
⟨*proof*⟩

lemmas *congruent-ctriangle-sas* = *congruent-ctriangleD*[*OF congruent-ctriangleI-sas*]

lemma *congruent-ctriangleI-aas*:
assumes *h1*:*angle-c a1 b1 c1* = *angle-c a2 b2 c2*

assumes *h2*:*angle-c b1 c1 a1* = *angle-c b2 c2 a2*
assumes *h3*:*cdist a1 b1* = *cdist a2 b2*
assumes *h4*:¬*collinear a1 b1 c1* ¬*collinear a2 b2 c2*
shows *congruent-ctriangle a1 b1 c1 a2 b2 c2*
⟨*proof*⟩

lemmas *congruent-ctriangle-aas* = *congruent-ctriangleD*[*OF congruent-ctriangleI-aas*]

lemma *congruent-ctriangleI-asa*:
assumes *angle-c a1 b1 c1* = *angle-c a2 b2 c2*
assumes *cdist a1 b1* = *cdist a2 b2*
assumes *h0*:*angle-c b1 a1 c1* = *angle-c b2 a2 c2*
assumes *h4*:¬*collinear a1 b1 c1* ¬*collinear a2 b2 c2*
shows *congruent-ctriangle a1 b1 c1 a2 b2 c2*
⟨*proof*⟩

lemmas *congruent-ctriangle-asa* = *congruent-ctriangleD*[*OF congruent-ctriangleI-asa*]

lemma *orientation-respect-letter-order*:
assumes *angle-c a b c* = *angle-c b a c* ¬ *collinear a b c*
shows *False*
⟨*proof*⟩

lemma *isoscele-iff-pr-cis-qr*:
assumes *h'*:‹*q*≠*r*›
shows ‹(*cdist q r* = *cdist p r*) ⟷ (*p*−*r*) = *cis*(*angle-c q r p*)∗(*q*−*r*)›
⟨*proof*⟩

lemma *equilateral-imp-pi3*:
assumes ‹*q*≠*r*› *cdist q r* = *cdist p r cdist p r* = *cdist p q*
shows ⌊(*angle-c q r p*)⌋ = *pi*/*3* ∨ ⌊(*angle-c q r p*)⌋ = −*pi*/*3*
(*angle-c q r p*) = (*angle-c p q r*) ∧ (*angle-c q r p*) = (*angle-c r p q*)
⟨*proof*⟩

lemma *isosceles-triangle-converse*:
assumes *angle-c a b c* = *angle-c c a b* ¬*collinear a b c*
shows *dist a c* = *dist b c*
⟨*proof*⟩

lemma *pi3-imp-equilateral*:
assumes ‹*q*≠*r*› ‹*p*≠*q*› ‹*r*≠*p*›
and ‹(*angle-c q p r*) = *pi*/*3* ∨ (*angle-c q p r*) = −*pi*/*3*›
and ‹(*angle-c q p r*) = (*angle-c r q p*)›

and ‹(*angle-c q p r*) = (*angle-c p r q*)›
shows ‹*cdist p r* = *cdist q r* ∧ *cdist p r* = *cdist p q*›
⟨*proof*⟩

lemma *pi3-isoscele-imp-equilateral*:
assumes ‹*q*≠*r*› ‹*p*≠*q*› *cdist q r* = *cdist p r*
and ⌊(*angle-c q p r*)⌋ = *pi*/*3* ∨ ⌊(*angle-c q p r*)⌋ = −*pi*/*3*
shows ‹*cdist p q* = *cdist r q*›
⟨*proof*⟩

lemma *pi3-isoscele-imp-equilateral′*:
assumes ‹*q*≠*r*› ‹*p*≠*q*› *cdist q r* = *cdist p r*
and ⌊(*angle-c q p r*)⌋ = *pi*/*3* ∨ ⌊(*angle-c q p r*)⌋ = −*pi*/*3*
shows ‹*cdist p r* = *cdist p q*›
⟨*proof*⟩

lemma *equilateral-caracterization*:‹*q*≠*r* ⟹ (*cdist q r* = *cdist p r* ∧ *cdist p r* = *cdist p q*)
⟷ ((*p*−*r*) = *cis*(*pi*/*3*)∗(*q*−*r*) ∨ (*p*−*r*) = *cis*(−*pi*/*3*)∗(*q*−*r*))›
⟨*proof*⟩

lemmas *equilateral-imp-prcispi3* = *equilateral-caracterization*[*THEN iffD1*]

lemmas *prcispi3-imp-equilateral* = *equilateral-caracterization*[*THEN iffD2*]

lemma *equilateral-caracterization-neg*:
fixes *p q r*::*complex*
assumes *h1*:‹*p*≠*r*›
shows ‹(*cdist p r* = *cdist p q* ∧ *cdist p q* = *cdist q r* ∧ *angle-c q r p* = −*pi*/*3*)
⟷ *p* + *root3* ∗ *q* + *root3* ^ *2* ∗ *r* = *0*›
⟨*proof*⟩

end
theory *Complex-Axial-Symmetry*

imports *Complex-Angles Complex-Triangles*

begin

6 Axial symmetry in complex field

In the following we define the axial symmetry and prove basics properties.

context

fixes *z1 z2* :: *complex* **and** *α β* :: *complex*
assumes *neq0*:‹*z1≠z2*›
defines ‹*α ≡ (z1−z2)/(cnj z1 − cnj z2)*›
defines ‹*β ≡ (z2∗cnj z1 − z1∗cnj z2)/(cnj z1 − cnj z2)*›
begin

definition *axial-symmetry*::‹*complex ⇒ complex*› **where**
‹*axial-symmetry z ≡ cnj z∗(z1 − z2)/(cnj z1 − cnj z2) + (z2∗cnj z1 − z1∗cnj z2)/(cnj z1 − cnj z2)*›

lemma *norm-α-eq-1*:‹*cmod (α) = 1*›
⟨*proof*⟩

lemma *z1-inv*:‹*axial-symmetry z1 = z1*›
⟨*proof*⟩

lemma *z2-inv*:‹*axial-symmetry z2 = z2*›
⟨*proof*⟩

lemma *cmod-axial*:‹*cmod (axial-symmetry z − axial-symmetry z′) = cmod (α∗(cnj z − cnj z′))*›
⟨*proof*⟩

lemma *cmod-axial-inv*:‹*cmod (axial-symmetry z − axial-symmetry z′) = cmod (z − z′)*›
⟨*proof*⟩

lemma *axial-symmetry-dist1*:‹*cdist z1 z = cdist z1 (axial-symmetry z)*›
⟨*proof*⟩

lemma *axial-symmetry-dist2*:‹*cdist z2 z = dist z2 (axial-symmetry z)*›
⟨*proof*⟩

lemma *αβ*: ‹*α∗cnj β + β = 0*›
⟨*proof*⟩

lemma *involution-symmetry*:‹*axial-symmetry (axial-symmetry z) = z*›
⟨*proof*⟩

lemma *arg-α*:‹*Arg α = ⌊2∗Arg (z1−z2)⌋*›
⟨*proof*⟩

lemma *Arg-invol*:‹*Arg (axial-symmetry (axial-symmetry z)) = Arg z*›
⟨*proof*⟩

lemma *angle-sum-symmetry*:‹*z≠z1 ⟹ ⌊angle-c z z1 z2 + angle-c z2 z1 (axial-symmetry z)⌋ = angle-c z z1 (axial-symmetry z)*›
⟨*proof*⟩

lemma *angle-symmetry-eq-imp*:
assumes *h*:‹*z1*≠*z*› ‹*z2*≠*z*›
shows‹*angle-c z z1 z2* = − *angle-c* (*axial-symmetry z*) *z1 z2* ∨ *angle-c z z1 z2* = *angle-c* (*axial-symmetry z*) *z1 z2*›
⟨*proof*⟩

lemma *angle-symmetry*:
assumes *h*:‹*z1*≠*z*› ‹*z2*≠*z*›
and ‹*angle-c z z1 z2* = *angle-c* (*axial-symmetry z*) *z1 z2*›
shows ‹*z* = *axial-symmetry z*›
⟨*proof*⟩

lemma *line-is-inv*:‹*z*∈ *line z1 z2* ∧ *z*≠*z2* ∧ *z*≠*z1* ⟹ *z* = *axial-symmetry z*›
⟨*proof*⟩

lemma *dist-inv*:‹*cdist a b* = *cdist* (*axial-symmetry a*) (*axial-symmetry b*)›
⟨*proof*⟩

lemma *collinear-inv*: **assumes** ‹*collinear a b c*› **and** ‹*a* ≠ *b*∧ *b* ≠ *c* ∧ *c*≠*a*›
shows ‹*collinear* (*axial-symmetry a*) (*axial-symmetry b*) (*axial-symmetry c*)›
⟨*proof*⟩

lemma *axial-symmetry-eq-line*:‹*z*≠*z1* ∧ *z*≠*z2* ⟹ *z* = *axial-symmetry z* ⟹ *z* ∈ *line z1 z2*›
⟨*proof*⟩

lemma *angle-symmetry-eq*:
assumes *h*:‹*z1*≠*z*› ‹*z2*≠*z*› ‹*z*∉*line z1 z2*›
shows‹*angle-c z z1 z2* = − *angle-c* (*axial-symmetry z*) *z1 z2*›
⟨*proof*⟩

end
end
theory *Morley*

imports *Complex-Axial-Symmetry*

begin

7 Rotations

locale *complex-rotation* =
fixes *A*::*complex* **and** *ϑ*::*real*
begin

definition ‹*r z* = *A* + (*z*−*A*)∗*cis*(*ϑ*)›

lemma *cmod-inv-rotation*:‹*cmod* (*z*−*A*) = *cmod* (*r z* − *A*)›
⟨*proof*⟩

lemma *inner-ang*:‹*cos* (∠ *z1 z2*)∗(*cmod z1* ∗*cmod z2*) = *Re* (*innerprod z1 z2*)›
⟨*proof*⟩

lemma *ang-eq-cos-theta*:‹*z*≠*A* ⟹ *cos* (*angle-c z A* (*r z*)) = *cos* (ϑ)›
⟨*proof*⟩

lemma *cdist-dist*:‹*cdist* = *dist*›
⟨*proof*⟩

lemma *ang-eq-theta*:**assumes** *h*:‹*z*≠*A*› **shows** ‹*angle-c z A* (*r z*) = ⌊ϑ⌋›
⟨*proof*⟩

lemma *inj-r*:‹*inj r*›
⟨*proof*⟩

lemma *img-eqI*:‹*cdist A z1* = *cdist A z2* ∧ *angle-c z1 A z2* = ϑ ⟹ *z2* = *r z1*›
⟨*proof*⟩

lemma *r-id-iff*:‹⌊ϑ⌋ = *0* ⟷ *r* = *id*›
⟨*proof*⟩

end

lemma *axial-symmetry-eq*:‹*axial-symmetry B C P* = *axial-symmetry C B P*› **if** ‹*C*≠*B*› **for** *C B P*
⟨*proof*⟩

lemma *img-r-sym*:
assumes *h*:‹*z1* ≠ *z2*› ‹*z* ∉ *line z1 z2*›
shows ‹*axial-symmetry z1 z2 z* = *complex-rotation.r z1* (⌊*2*∗*angle-c z z1 z2*⌋) *z*›
⟨*proof*⟩

lemma *img-r-sym′*:
assumes *h*:‹*z1* ≠ *z2*› ‹*z*∉*line z1 z2*›
shows ‹*axial-symmetry z1 z2 z* = *complex-rotation.r z1* (⌊−*2*∗*angle-c z2 z1 z*⌋) *z*›
⟨*proof*⟩

lemma *equality-for-pqr*:
assumes *1*:‹(*a2*::*complex*)∗*a3*≠*1*› **and** *2*:‹⋀*z*. *h z* = *a3*∗*z* + *b3*› **and** *3*:‹⋀*z*. *g z* = *a2*∗*z* + *b2*› **and** *4*:‹*g* (*h z*) = *z*›
shows ‹*z* = (*a2*∗*b3* + *b2*)/(*1*−*a2*∗*a3*)›
⟨*proof*⟩

lemma *equality-for-comp*:

assumes *2*:‹⋀*z*. *h z* = (*a3*::*complex*)∗*z* + *b3*› **and** *3*:‹⋀*z*. *g z* = *a2*∗*z* + *b2*›
and *4*:‹⋀*z*. *f z* = *a1*∗*z* +*b1*›
shows ‹((*f*∘*f*∘*f*)∘(*g*∘*g*∘*g*)∘(*h*∘*h*∘*h*)) *z* = (*a1*∗*a2*∗*a3*)^*3*∗*z* +(*a1*^*2*+*a1*+*1*)∗*b1* +*a1*^*3*∗(*a2*^*2*+*a2*+*1*)∗*b2*
+*a1*^*3*∗*a2*^*3*∗(*a3*^*2*+*a3*+*1*)∗*b3* ›
⟨*proof*⟩

lemma *eq-translation-id*:
assumes ‹*h* = *complex-rotation.r A 0*› ‹*h B* = *B*›
shows ‹*h* = *id*›
⟨*proof*⟩

lemma *r-eqI*:
assumes ‹*A* = *B*› ‹*ϑ1* = *ϑ2*›
shows ‹*r A ϑ1* = *r B ϑ2*›
⟨*proof*⟩

lemma *r-eqI'*:
assumes ‹*A* = *B*› ‹*ϑ1* = *ϑ2*›
shows ‹*r A ϑ1 z* = *r B ϑ2 z*›
⟨*proof*⟩

lemma *composed-rotations-same-center*:
shows ‹(*complex-rotation.r A ϑ1* ∘ *complex-rotation.r A ϑ2*) = *complex-rotation.r A* (*ϑ1* + *ϑ2*)›
⟨*proof*⟩

lemma *composed-rotations*:
assumes *h*:‹⌊*ϑ1* + *ϑ2*⌋ ≠ *0*›
shows ‹(*complex-rotation.r A ϑ1* ∘ *complex-rotation.r B ϑ2*) =
complex-rotation.r ((*A*∗(*1*−*cis ϑ1*) + *B*∗*cis ϑ1*∗(*1*−*cis ϑ2*))/(*1*−*cis* (*ϑ1*+*ϑ2*))) (*ϑ1* + *ϑ2*)›
⟨*proof*⟩

lemma *composed-rotation-is-trans*:
assumes ‹⌊*ϑ1* + *ϑ2*⌋ = *0*›
shows ‹(*complex-rotation.r A ϑ1* ∘ *complex-rotation.r B ϑ2*) *z* = *z* + (*B* − *A*)∗(*cis*(*ϑ1*) − *1*)›
⟨*proof*⟩

8 Morley's theorem

We begin by proving the Morley's theorem in the case where angles are positives then using the congruence between two triangles with the same angles only not of the same sign we prove Morley's theorem when angles are negatives.

We then proceed to conclude because in a triangle either angles are all negatives or all the angles are positives depending on orientation.

theorem *Morley-pos*:
 assumes‹¬*collinear A B C*›
  ‹*angle-c A B R* = *angle-c A B C* / *3*› (**is** ‹*?abr* = *?abc*›)
  angle-c B A R = *angle-c B A C* / *3* (**is** ‹*?bar* = *?α*›)
  angle-c B C P = *angle-c B C A* / *3* (**is** ‹*?bcp* = *?bca*›)
  angle-c C B P = *angle-c C B A* / *3* (**is** ‹*?cbp* = *?β*›)
  angle-c C A Q = *angle-c C A B* / *3* (**is** ‹*?caq* = *?cab*›)
  angle-c A C Q = *angle-c A C B* / *3* (**is** ‹*?acq* = *?γ*›)
 and *hhh*:‹⌊*angle-c B A C* / *3*+*angle-c C B A* / *3*+*angle-c A C B* / *3*⌋ = *pi*/*3*›
 shows ‹*cdist R P* = *cdist P Q* ∧ *cdist Q R* = *cdist P Q*›
⟨*proof*⟩

theorem *Morley-neg*:
 assumes‹¬*collinear A B C*›
  ‹*angle-c A B R* = *angle-c A B C* / *3*› (**is** ‹*?abr* = *?abc*›)
  angle-c B A R = *angle-c B A C* / *3* (**is** ‹*?bar* = *?α*›)
  angle-c B C P = *angle-c B C A* / *3* (**is** ‹*?bcp* = *?bca*›)
  angle-c C B P = *angle-c C B A* / *3* (**is** ‹*?cbp* = *?β*›)
  angle-c C A Q = *angle-c C A B* / *3* (**is** ‹*?caq* = *?cab*›)
  angle-c A C Q = *angle-c A C B* / *3* (**is** ‹*?acq* = *?γ*›)
 and *hhh*:‹⌊*angle-c B A C* / *3*+*angle-c C B A* / *3*+*angle-c A C B* / *3*⌋ = −*pi*/*3*›
 shows ‹*cdist R P* = *cdist P Q* ∧ *cdist Q R* = *cdist P Q*›
⟨*proof*⟩

theorem *Morley*:
 assumes‹¬*collinear A B C*›
  ‹*angle-c A B R* = *angle-c A B C* / *3*› (**is** ‹*?abr* = *?abc*›)
  angle-c B A R = *angle-c B A C* / *3* (**is** ‹*?bar* = *?α*›)
  angle-c B C P = *angle-c B C A* / *3* (**is** ‹*?bcp* = *?bca*›)
  angle-c C B P = *angle-c C B A* / *3* (**is** ‹*?cbp* = *?β*›)
  angle-c C A Q = *angle-c C A B* / *3* (**is** ‹*?caq* = *?cab*›)
  angle-c A C Q = *angle-c A C B* / *3* (**is** ‹*?acq* = *?γ*›)
 shows ‹*cdist R P* = *cdist P Q* ∧ *cdist Q R* = *cdist P Q*›
⟨*proof*⟩

end

References

[1] Manuel Eberl *Basic Geometric Properties of Triangles*, Archive of Formal Proofs, December 2015 https://isa-afp.org/entries/Triangle.html

[2] Morley theorem http://denisevellachemla.eu/Alain-Connes-Theoreme-Morley.pdf.

[3] Wiedijk's catalogue "Formalizing 100 Theorems" https://www.cs.ru.nl/~freek/100/,Itappearsatposition121....