Conway's Circle Theorem in Isabelle/HOL

Arthur Freitas Ramos David Barros Hulak
Ruy J. G. B. de Queiroz

July 15, 2026

Abstract

We formalize Conway's circle theorem in Isabelle/HOL. For a non-degenerate Euclidean triangle ABC with side lengths $a = |BC|$, $b = |CA|$, and $c = |AB|$, extend the two side-lines meeting at each vertex past that vertex by the length of the side opposite that vertex: at A both adjacent sides are extended by a, at B by b, and at C by c. The six resulting endpoints lie on one circle. Its center is the incenter I of the triangle, and its radius is

$$\sqrt{r^2 + s^2},$$

where r is the inradius and $s = (a + b + c)/2$ is the semiperimeter.

We give the result in three equivalent forms: as six explicit distance equations, as a subset relation $\subseteq$ into the corresponding sphere, and as a single bounded-universal statement over the six-point set. A separate construction-correctness theorem records that each Conway point lies on the intended side line and at the declared extension length from the adjacent vertex.

AI assistance was used for proof engineering.

Contents

1 Introduction **2**

2 Definitions **2**

3 Proof Sketch **2**

4 Historical Remark **3**

4.1 Wrapper formulations 21

1 Introduction

Conway's construction is a compact Euclidean configuration: a triangle determines three side lengths, those lengths determine six points on the three side lines, and the resulting endpoints unexpectedly form a circle. The theorem is often presented as a visual fact about equal segments; the formal proof has to make explicit which directed lines are used, which extensions are external to the triangle, and how the metric computation is organized.

The development follows the elementary proof. Each constructed point lies on one of the three side lines. The perpendicular from the incenter I to that side has length r, and the signed distance from the foot of the perpendicular to either constructed endpoint is s. The Pythagorean theorem therefore gives the common distance

$$\sqrt{r^2 + s^2}$$

from I to all six endpoints.

2 Definitions

The formal statement works in the Euclidean plane over real coordinates. The vertices A, B, and C are assumed not to be collinear. The side lengths are named in the usual way:

$$\begin{aligned} a &= |BC|, \qquad & b = |CA|, \quad c = |AB|, \\ s &= \frac{a+b+c}{2}. \end{aligned}$$

The incenter I is the point whose perpendicular distance to each side line is the common inradius r.

For the construction, each side line contributes two points. On the line AB, the point A_c is beyond A with $|AA_c| = a$, and the point B_c is beyond B with $|BB_c| = b$. Cyclically, the line AC gives A_b and C_b, and the line BC gives B_a and C_a. These prose names are only a guide to the geometric objects; the precise predicates and uniqueness facts are those rendered from the Isabelle theory below.

3 Proof Sketch

Let T_{AB} be the touch point of the incircle with the side AB. The standard tangent-length identities give

$$|AT_{AB}| = s - a, \qquad |BT_{AB}| = s - b.$$

Since A_c lies past A by length a, the directed distance from T_{AB} to A_c has absolute value s; similarly the directed distance from T_{AB} to B_c has absolute value s. The segment IT_{AB} is perpendicular to AB and has length r, so the two right triangles with hypotenuses IA_c and IB_c both have legs r and s. Thus

$$|IA_c| = |IB_c| = \sqrt{r^2 + s^2}.$$

The same argument on the side lines AC and BC proves the identical distance formula for A_b, C_b, B_a, and C_a. Hence all six constructed points lie on the circle with center I and radius $\sqrt{r^2 + s^2}$.

4 Historical Remark

The result is part of the informal triangle-geometry tradition associated with John H. Conway, and public expositions record the construction under his name [3]. We do not attempt a precise priority claim here. Later conic work by Evans fits the theorem into the broader triangle-center literature [2], and García Capitán subsequently showed that the same six-point pattern belongs to a family of conics, with the Conway circle recovered when the chosen point is the incenter [1]. Honsberger's collection gives related Euclidean-geometry background [4].

theory *Conway-Circle*
  imports *HOL−Analysis.Analysis*
begin

definition *triangle-semiperimeter* :: $real \,\hat{}\, 2 \Rightarrow real \,\hat{}\, 2 \Rightarrow real \,\hat{}\, 2 \Rightarrow real$
  where *triangle-semiperimeter* $A\ B\ C =$
    $(dist\ B\ C + dist\ C\ A + dist\ A\ B)\ /\ 2$

definition *triangle-incenter* :: $real \,\hat{}\, 2 \Rightarrow real \,\hat{}\, 2 \Rightarrow real \,\hat{}\, 2 \Rightarrow real \,\hat{}\, 2$
  where *triangle-incenter* $A\ B\ C =$
    $(let\ a = dist\ B\ C;\ b = dist\ C\ A;\ c = dist\ A\ B\ in$
      $inverse\ (a + b + c) *_R (a *_R A + b *_R B + c *_R C))$

definition *conway-point-past-first* :: $'a::real\text{-}normed\text{-}vector \Rightarrow 'a \Rightarrow real \Rightarrow 'a$
  where *conway-point-past-first* $X\ Y\ d = X + (d\ /\ dist\ X\ Y) *_R (X - Y)$

definition *conway-point-past-second* :: $'a::real\text{-}normed\text{-}vector \Rightarrow 'a \Rightarrow real \Rightarrow 'a$
  where *conway-point-past-second* $X\ Y\ d = Y + (d\ /\ dist\ X\ Y) *_R (Y - X)$

definition *side-contact-point* :: $'a::real\text{-}normed\text{-}vector \Rightarrow 'a \Rightarrow real \Rightarrow real \Rightarrow 'a$
  where *side-contact-point* $X\ Y\ p\ q =$
    $(let\ L = dist\ X\ Y;\ s = (p + q + L)\ /\ 2\ in\ X + ((s - p)\ /\ L) *_R (Y - X))$

definition *conway-A-c* :: $real \,\hat{}\, 2 \Rightarrow real \,\hat{}\, 2 \Rightarrow real \,\hat{}\, 2 \Rightarrow real \,\hat{}\, 2$
  where *conway-A-c* $A\ B\ C =$ *conway-point-past-first* $A\ B\ (dist\ B\ C)$

definition *conway-B-c* :: *real* ^ *2* ⇒ *real* ^ *2* ⇒ *real* ^ *2* ⇒ *real* ^ *2*
  where *conway-B-c A B C* = *conway-point-past-second A B* (*dist C A*)

definition *conway-A-b* :: *real* ^ *2* ⇒ *real* ^ *2* ⇒ *real* ^ *2* ⇒ *real* ^ *2*
  where *conway-A-b A B C* = *conway-point-past-first A C* (*dist B C*)

definition *conway-C-b* :: *real* ^ *2* ⇒ *real* ^ *2* ⇒ *real* ^ *2* ⇒ *real* ^ *2*
  where *conway-C-b A B C* = *conway-point-past-second A C* (*dist A B*)

definition *conway-B-a* :: *real* ^ *2* ⇒ *real* ^ *2* ⇒ *real* ^ *2* ⇒ *real* ^ *2*
  where *conway-B-a A B C* = *conway-point-past-first B C* (*dist C A*)

definition *conway-C-a* :: *real* ^ *2* ⇒ *real* ^ *2* ⇒ *real* ^ *2* ⇒ *real* ^ *2*
  where *conway-C-a A B C* = *conway-point-past-second B C* (*dist A B*)

definition *contact-AB* :: *real* ^ *2* ⇒ *real* ^ *2* ⇒ *real* ^ *2* ⇒ *real* ^ *2*
  where *contact-AB A B C* = *side-contact-point A B* (*dist B C*) (*dist C A*)

definition *contact-AC* :: *real* ^ *2* ⇒ *real* ^ *2* ⇒ *real* ^ *2* ⇒ *real* ^ *2*
  where *contact-AC A B C* = *side-contact-point A C* (*dist B C*) (*dist A B*)

definition *contact-BC* :: *real* ^ *2* ⇒ *real* ^ *2* ⇒ *real* ^ *2* ⇒ *real* ^ *2*
  where *contact-BC A B C* = *side-contact-point B C* (*dist C A*) (*dist A B*)

definition *triangle-inradius* :: *real* ^ *2* ⇒ *real* ^ *2* ⇒ *real* ^ *2* ⇒ *real*
  where *triangle-inradius A B C* =
    dist (*triangle-incenter A B C*) (*contact-AB A B C*)

lemma *noncollinear-triangle-distinct*:
  assumes ¬ *collinear* {*A*, *B*, *C*}
  shows *A* ≠ *B A* ≠ *C B* ≠ *C*
proof −
  show *A* ≠ *B*
  proof
    assume *A* = *B*
    then have {*A*, *B*, *C*} = {*A*, *C*}
      by *auto*
    then have *collinear* {*A*, *B*, *C*}
      by *simp*
    with *assms* **show** *False*
      by *contradiction*
  qed
  show *A* ≠ *C*
  proof
    assume *A* = *C*
    then have {*A*, *B*, *C*} = {*B*, *C*}
      by *auto*
    then have *collinear* {*A*, *B*, *C*}
      by *simp*

```
    with assms show False
      by contradiction
  qed
  show B ≠ C
  proof
    assume B = C
    then have {A, B, C} = {A, C}
      by auto
    then have collinear {A, B, C}
      by simp
    with assms show False
      by contradiction
  qed
qed

lemma inner-side-vectors:
  fixes X Y Z :: 'a::real-inner
  shows (Z − X) · (Y − X) =
    ((dist Z X)^2 + (dist X Y)^2 − (dist Y Z)^2) / 2
proof −
  have (Z − X) · (Y − X) =
    (((norm (Z − X))^2 + (norm (Y − X))^2) − (norm ((Z − X) − (Y − X)))^2)
/ 2
    by (rule dot-norm-neg)
  also have ... = ((dist Z X)^2 + (dist X Y)^2 − (dist Y Z)^2) / 2
    by (simp add: dist-norm norm-minus-commute)
  finally show ?thesis .
qed

lemma weighted-incenter-side-contact-orthogonal:
  fixes X Y Z :: 'a::real-inner
  assumes X ≠ Y
  defines a ≡ dist Y Z
  defines b ≡ dist Z X
  defines c ≡ dist X Y
  defines S ≡ a + b + c
  defines I ≡ inverse S *R (a *R X + b *R Y + c *R Z)
  defines T ≡ side-contact-point X Y a b
  shows orthogonal (I − T) (Y − X)
proof −
  have cpos: 0 < c
    using assms unfolding c-def by simp
  have a-nonneg: 0 ≤ a
    unfolding a-def by simp
  have b-nonneg: 0 ≤ b
    unfolding b-def by simp
  have Spos: 0 < S
    using a-nonneg b-nonneg cpos unfolding S-def by linarith
  define u where u = Y − X
```

define v **where** $v = Z - X$
have *incenter-translate*:
$I - X = \mathit{inverse}\ S *_R (b *_R u + c *_R v)$
proof −
have *numerator*:
$a *_R X + b *_R Y + c *_R Z =$
$S *_R X + (b *_R (Y - X) + c *_R (Z - X))$
unfolding *S-def* **by** (*simp add*: *algebra-simps scaleR-diff-right*)
have $I = X + \mathit{inverse}\ S *_R (b *_R (Y - X) + c *_R (Z - X))$
unfolding *I-def numerator*
using *Spos* **by** (*simp add*: *scaleR-add-right*)
then show *?thesis*
unfolding *u-def v-def* **by** *simp*
qed
have *contact-translate*:
$T - X = ((b + c - a) / (2 * c)) *_R u$
unfolding *T-def side-contact-point-def a-def b-def c-def Let-def*
by (*simp add*: *field-simps u-def*)
have *side-square*: $u \cdot u = c^2$
proof −
have $(Y - X) \cdot (Y - X) = (\mathit{norm}\ (Y - X))^2$
by (*simp add*: *power2-norm-eq-inner*)
also have ... $= c^2$
unfolding *c-def* **by** (*simp add*: *dist-norm norm-minus-commute*)
finally show *?thesis*
unfolding *u-def* **.**
qed
have *side-cross*: $v \cdot u = (b^2 + c^2 - a^2) / 2$
unfolding *u-def v-def a-def b-def c-def* **by** (*rule inner-side-vectors*)
have $(I - T) \cdot (Y - X) =$
$((I - X) - (T - X)) \cdot u$
unfolding *u-def* **by** *simp*
also have ... $=$
$(\mathit{inverse}\ S *_R (b *_R u + c *_R v) -$
$((b + c - a) / (2 * c)) *_R u) \cdot u$
unfolding *incenter-translate contact-translate* **by** *simp*
also have ... $=$
$\mathit{inverse}\ S * (b * c^2 + c * ((b^2 + c^2 - a^2) / 2)) -$
$((b + c - a) / (2 * c)) * c^2$
by (*simp add*: *inner-simps side-square side-cross*)
also have ... $= 0$
using *cpos Spos*
by (*simp add*: *field-simps power2-eq-square S-def*)
finally have $(I - T) \cdot (Y - X) = 0$ **.**
then show *?thesis*
unfolding *orthogonal-def* **by** *simp*
qed

lemma *incenter-contact-dist-sq-scalar*:

```
  fixes a b c S :: real
  assumes c ≠ 0
    and S ≠ 0
    and S = a + b + c
  shows
    (inverse S * b − (b + c − a) / (2 * c))² * c² +
      (inverse S * c)² * b² +
      2 * (inverse S * b − (b + c − a) / (2 * c)) * (inverse S * c) *
        ((b² + c² − a²) / 2) =
    ((b + c − a) * (a + c − b) * (a + b − c)) / (4 * S)
proof −
  define D where D = a² − b² − c²
  have alpha-eq:
    inverse S * b − (b + c − a) / (2 * c) = D / (2 * c * S)
    using assms unfolding D-def
    by (simp add: field-simps power2-eq-square)
  have beta-eq: inverse S * c = c / S
    by (simp add: divide-inverse mult.commute)
  have cross-eq: (b² + c² − a²) / 2 = − D / 2
    unfolding D-def by simp
  have
    (inverse S * b − (b + c − a) / (2 * c))² * c² +
      (inverse S * c)² * b² +
      2 * (inverse S * b − (b + c − a) / (2 * c)) * (inverse S * c) *
        ((b² + c² − a²) / 2) =
    (D / (2 * c * S))² * c² + (c / S)² * b² +
      2 * (D / (2 * c * S)) * (c / S) * (− D / 2)
    by (simp add: alpha-eq beta-eq cross-eq)
  also have ... = (4 * b² * c² − D²) / (4 * S²)
    using assms(1,2) by (simp add: field-simps power2-eq-square)
  also have ... = ((2 * b * c − D) * (2 * b * c + D)) / (4 * S²)
    by (simp add: power2-eq-square algebra-simps)
  also have ... =
      (((b + c)² − a²) * (a² − (b − c)²)) / (4 * S²)
    unfolding D-def by (simp add: power2-eq-square algebra-simps)
  also have ... =
      ((S * (b + c − a)) * ((a + c − b) * (a + b − c))) / (4 * S²)
    using assms by (simp add: power2-eq-square algebra-simps)
  also have ... =
      ((b + c − a) * (a + c − b) * (a + b − c)) / (4 * S)
    using assms(2) by (simp add: field-simps power2-eq-square)
  finally show ?thesis .
qed

lemma weighted-incenter-side-contact-dist-sq:
  fixes X Y Z :: 'a::real-inner
  assumes X ≠ Y
  defines a ≡ dist Y Z
  defines b ≡ dist Z X
```

```
  defines c ≡ dist X Y
  defines S ≡ a + b + c
  defines I ≡ inverse S *R (a *R X + b *R Y + c *R Z)
  defines T ≡ side-contact-point X Y a b
  shows (dist I T)^2 =
    ((b + c − a) * (a + c − b) * (a + b − c)) / (4 * S)
proof −
  have cpos: 0 < c
    using assms unfolding c-def by simp
  have a-nonneg: 0 ≤ a
    unfolding a-def by simp
  have b-nonneg: 0 ≤ b
    unfolding b-def by simp
  have Spos: 0 < S
    using a-nonneg b-nonneg cpos unfolding S-def by linarith
  define u where u = Y − X
  define v where v = Z − X
  have incenter-translate:
    I − X = inverse S *R (b *R u + c *R v)
  proof −
    have numerator:
      a *R X + b *R Y + c *R Z =
        S *R X + (b *R (Y − X) + c *R (Z − X))
      unfolding S-def by (simp add: algebra-simps scaleR-diff-right)
    have I = X + inverse S *R (b *R (Y − X) + c *R (Z − X))
      unfolding I-def numerator
      using Spos by (simp add: scaleR-add-right)
    then show ?thesis
      unfolding u-def v-def by simp
  qed
  have contact-translate:
    T − X = ((b + c − a) / (2 * c)) *R u
    unfolding T-def side-contact-point-def a-def b-def c-def Let-def
    by (simp add: field-simps u-def)
  have diff-translate:
    I − T =
      inverse S *R (b *R u + c *R v) −
      ((b + c − a) / (2 * c)) *R u
  proof −
    have I − T = (I − X) − (T − X)
      by simp
    also have ... =
      inverse S *R (b *R u + c *R v) −
      ((b + c − a) / (2 * c)) *R u
      unfolding incenter-translate contact-translate ..
    finally show ?thesis .
  qed
  have side-square: u · u = c^2
  proof −
```

have $(Y - X) \cdot (Y - X) = (norm\ (Y - X))^2$
by (*simp add*: *power2-norm-eq-inner*)
also have ... $= c^2$
unfolding *c-def* **by** (*simp add*: *dist-norm norm-minus-commute*)
finally show *?thesis*
unfolding *u-def* **.**
qed
have *vertex-square*: $v \cdot v = b^2$
proof −
have $(Z - X) \cdot (Z - X) = (norm\ (Z - X))^2$
by (*simp add*: *power2-norm-eq-inner*)
also have ... $= b^2$
unfolding *b-def* **by** (*simp add*: *dist-norm norm-minus-commute*)
finally show *?thesis*
unfolding *v-def* **.**
qed
have *side-cross*: $v \cdot u = (b^2 + c^2 - a^2) / 2$
unfolding *u-def v-def a-def b-def c-def* **by** (*rule inner-side-vectors*)
have *side-cross'*: $u \cdot v = (b^2 + c^2 - a^2) / 2$
using *side-cross* **by** (*simp add*: *inner-commute*)
define *alpha* **where** $alpha = inverse\ S * b - ((b + c - a) / (2 * c))$
define *beta* **where** $beta = inverse\ S * c$
have *linear-diff*:
$inverse\ S *_R (b *_R u + c *_R v) -$
$((b + c - a) / (2 * c)) *_R u =$
$alpha *_R u + beta *_R v$
unfolding *alpha-def beta-def*
by (*simp add*: *algebra-simps scaleR-add-right scaleR-diff-left*)
have $(dist\ I\ T)^2 = (I - T) \cdot (I - T)$
by (*simp add*: *dist-norm power2-norm-eq-inner*)
also have ... $=$
$(inverse\ S *_R (b *_R u + c *_R v) -$
$((b + c - a) / (2 * c)) *_R u) \cdot$
$(inverse\ S *_R (b *_R u + c *_R v) -$
$((b + c - a) / (2 * c)) *_R u)$
unfolding *diff-translate* **..**
also have ... $=$
$(alpha *_R u + beta *_R v) \cdot (alpha *_R u + beta *_R v)$
unfolding *linear-diff* **..**
also have ... $=$
$alpha^2 * c^2 + beta^2 * b^2 +$
$2 * alpha * beta * ((b^2 + c^2 - a^2) / 2)$
by (*simp add*: *inner-simps side-square vertex-square side-cross side-cross'*
power2-eq-square algebra-simps) *algebra*
also have ... $=$
$(inverse\ S * b - ((b + c - a) / (2 * c)))^2 * c^2 +$
$(inverse\ S * c)^2 * b^2 +$
$2 * (inverse\ S * b - ((b + c - a) / (2 * c))) *$
$(inverse\ S * c) * ((b^2 + c^2 - a^2) / 2)$

```
    unfolding alpha-def beta-def ..
  also have ... =
      ((b + c − a) ∗ (a + c − b) ∗ (a + b − c)) / (4 ∗ S)
  proof (rule incenter-contact-dist-sq-scalar)
    show c ≠ 0
      using cpos by simp
    show S ≠ 0
      using Spos by simp
    show S = a + b + c
      unfolding S-def ..
  qed
  finally show ?thesis .
qed

lemma triangle-incenter-contact-AB-orthogonal:
  fixes A B C :: real ^ 2
  assumes A ≠ B
  shows orthogonal (triangle-incenter A B C − contact-AB A B C) (B − A)
  using weighted-incenter-side-contact-orthogonal[OF assms, of C]
  unfolding triangle-incenter-def contact-AB-def Let-def
  by simp

lemma triangle-incenter-contact-AC-orthogonal:
  fixes A B C :: real ^ 2
  assumes A ≠ C
  shows orthogonal (triangle-incenter A B C − contact-AC A B C) (C − A)
  using weighted-incenter-side-contact-orthogonal[OF assms, of B]
  unfolding triangle-incenter-def contact-AC-def Let-def
  by (simp add: algebra-simps dist-commute)

lemma triangle-incenter-contact-BC-orthogonal:
  fixes A B C :: real ^ 2
  assumes B ≠ C
  shows orthogonal (triangle-incenter A B C − contact-BC A B C) (C − B)
  using weighted-incenter-side-contact-orthogonal[OF assms, of A]
  unfolding triangle-incenter-def contact-BC-def Let-def
  by (simp add: algebra-simps dist-commute)

lemma triangle-inradius-nonneg [simp]: 0 ≤ triangle-inradius A B C
  unfolding triangle-inradius-def by simp

lemma triangle-incenter-contact-AB-dist:
  dist (triangle-incenter A B C) (contact-AB A B C) = triangle-inradius A B C
  unfolding triangle-inradius-def ..

lemma triangle-incenter-contact-AC-dist:
  fixes A B C :: real ^ 2
  assumes triangle: ¬ collinear {A, B, C}
  shows dist (triangle-incenter A B C) (contact-AC A B C) = triangle-inradius A
```

B C
proof −
  let *?I* = *triangle-incenter A B C*
  let *?a* = *dist B C*
  let *?b* = *dist C A*
  let *?c* = *dist A B*
  let *?H* = ((*?b* + *?c* − *?a*) ∗ (*?a* + *?c* − *?b*) ∗ (*?a* + *?b* − *?c*)) /
    (*4* ∗ (*?a* + *?b* + *?c*))
  have *sides*: *A* ≠ *B A* ≠ *C*
    using *noncollinear-triangle-distinct*[*OF triangle*] **by** *blast+*
  have *AB-sq*: (*dist ?I* (*contact-AB A B C*))^2 = *?H*
    using *weighted-incenter-side-contact-dist-sq*[*OF sides*(*1*), *of C*]
    unfolding *triangle-incenter-def contact-AB-def Let-def*
    by (*simp add*: *dist-commute algebra-simps*)
  have *AC-sq*: (*dist ?I* (*contact-AC A B C*))^2 = *?H*
    using *weighted-incenter-side-contact-dist-sq*[*OF sides*(*2*), *of B*]
    unfolding *triangle-incenter-def contact-AC-def Let-def*
    by (*simp add*: *dist-commute algebra-simps*)
  show *?thesis*
    unfolding *triangle-inradius-def*
  proof (*rule power2-eq-imp-eq*)
    show (*dist ?I* (*contact-AC A B C*))^2 =
        (*dist ?I* (*contact-AB A B C*))^2
      using *AC-sq AB-sq* **by** *simp*
    show *0* ≤ *dist ?I* (*contact-AC A B C*)
      by *simp*
    show *0* ≤ *dist ?I* (*contact-AB A B C*)
      by *simp*
  qed
qed

lemma *triangle-incenter-contact-BC-dist*:
  fixes *A B C* :: *real* ^ *2*
  assumes *triangle*: ¬ *collinear* {*A*, *B*, *C*}
  shows *dist* (*triangle-incenter A B C*) (*contact-BC A B C*) = *triangle-inradius A B C*
proof −
  let *?I* = *triangle-incenter A B C*
  let *?a* = *dist B C*
  let *?b* = *dist C A*
  let *?c* = *dist A B*
  let *?H* = ((*?b* + *?c* − *?a*) ∗ (*?a* + *?c* − *?b*) ∗ (*?a* + *?b* − *?c*)) /
    (*4* ∗ (*?a* + *?b* + *?c*))
  have *sides*: *A* ≠ *B B* ≠ *C*
    using *noncollinear-triangle-distinct*[*OF triangle*] **by** *blast+*
  have *AB-sq*: (*dist ?I* (*contact-AB A B C*))^2 = *?H*
    using *weighted-incenter-side-contact-dist-sq*[*OF sides*(*1*), *of C*]
    unfolding *triangle-incenter-def contact-AB-def Let-def*
    by (*simp add*: *dist-commute algebra-simps*)

```
  have BC-sq: (dist ?I (contact-BC A B C))² = ?H
    using weighted-incenter-side-contact-dist-sq[OF sides(2), of A]
    unfolding triangle-incenter-def contact-BC-def Let-def
    by (simp add: dist-commute algebra-simps)
  show ?thesis
    unfolding triangle-inradius-def
  proof (rule power2-eq-imp-eq)
    show (dist ?I (contact-BC A B C))² =
        (dist ?I (contact-AB A B C))²
      using BC-sq AB-sq by simp
    show 0 ≤ dist ?I (contact-BC A B C)
      by simp
    show 0 ≤ dist ?I (contact-AB A B C)
      by simp
  qed
qed

lemma triangle-incenter-contacts-dist:
  fixes A B C :: real ^ 2
  assumes ¬ collinear {A, B, C}
  shows
    dist (triangle-incenter A B C) (contact-AB A B C) = triangle-inradius A B C
    dist (triangle-incenter A B C) (contact-AC A B C) = triangle-inradius A B C
    dist (triangle-incenter A B C) (contact-BC A B C) = triangle-inradius A B C
proof –
  show dist (triangle-incenter A B C) (contact-AB A B C) = triangle-inradius A
B C
    by (rule triangle-incenter-contact-AB-dist)
  show dist (triangle-incenter A B C) (contact-AC A B C) = triangle-inradius A
B C
    by (rule triangle-incenter-contact-AC-dist[OF assms])
  show dist (triangle-incenter A B C) (contact-BC A B C) = triangle-inradius A
B C
    by (rule triangle-incenter-contact-BC-dist[OF assms])
qed

lemma triangle-incenter-contacts-orthogonal:
  fixes A B C :: real ^ 2
  assumes ¬ collinear {A, B, C}
  shows
    orthogonal (triangle-incenter A B C − contact-AB A B C) (B − A)
    orthogonal (triangle-incenter A B C − contact-AC A B C) (C − A)
    orthogonal (triangle-incenter A B C − contact-BC A B C) (C − B)
proof –
  have sides: A ≠ B A ≠ C B ≠ C
    using noncollinear-triangle-distinct[OF assms] by blast+
  show orthogonal (triangle-incenter A B C − contact-AB A B C) (B − A)
    by (rule triangle-incenter-contact-AB-orthogonal[OF sides(1)])
  show orthogonal (triangle-incenter A B C − contact-AC A B C) (C − A)
```

by (*rule triangle-incenter-contact-AC-orthogonal*[*OF sides*(*2*)])
show *orthogonal* (*triangle-incenter A B C* − *contact-BC A B C*) (*C* − *B*)
by (*rule triangle-incenter-contact-BC-orthogonal*[*OF sides*(*3*)])
qed

lemma *dist-conway-point-past-first*:
fixes *X Y* :: ′*a*::*real-normed-vector*
assumes $X \neq Y$ $0 \leq d$
shows *dist X* (*conway-point-past-first X Y d*) = *d*
proof −
have *0* < *dist X Y*
using *assms* **by** *simp*
then show *?thesis*
using *assms*
unfolding *conway-point-past-first-def*
by (*simp add*: *dist-norm norm-minus-commute*)
qed

lemma *dist-conway-point-past-second*:
fixes *X Y* :: ′*a*::*real-normed-vector*
assumes $X \neq Y$ $0 \leq d$
shows *dist Y* (*conway-point-past-second X Y d*) = *d*
proof −
have *0* < *dist X Y*
using *assms* **by** *simp*
then show *?thesis*
using *assms*
unfolding *conway-point-past-second-def*
by (*simp add*: *dist-norm norm-minus-commute*)
qed

lemma *conway-point-past-first-contact*:
fixes *X Y* :: ′*a*::*real-normed-vector*
assumes $X \neq Y$
shows *conway-point-past-first X Y p* =
side-contact-point X Y p q − (((*p* + *q* + *dist X Y*) / *2*) / *dist X Y*) $*_R$ (*Y* −
X)
proof −
let *?L* = *dist X Y*
let *?s* = (*p* + *q* + *?L*) / *2*
have *Lpos*: *0* < *?L*
using *assms* **by** *simp*
have *scalar*: ((*?s* − *p*) / *?L* − *?s* / *?L*) = − (*p* / *?L*)
using *Lpos* **by** (*simp add*: *field-simps*)
have *side-contact-point X Y p q* − (*?s* / *?L*) $*_R$ (*Y* − *X*) =
X + ((*?s* − *p*) / *?L*) $*_R$ (*Y* − *X*) − (*?s* / *?L*) $*_R$ (*Y* − *X*)
unfolding *side-contact-point-def Let-def* **by** *simp*
also have ... = *X* + (((*?s* − *p*) / *?L* − *?s* / *?L*) $*_R$ (*Y* − *X*))
by (*simp add*: *scaleR-diff-left* [*symmetric*])

also have ... = X + (−(p / $?L$) $*_R$ (Y − X))
  using *scalar* **by** *simp*
also have ... = X + (p / $?L$) $*_R$ (X − Y)
  by (*simp add*: *scaleR-right-diff-distrib*)
also have ... = *conway-point-past-first* X Y p
  unfolding *conway-point-past-first-def* **..**
finally show *?thesis*
  by (*rule sym*)
qed

lemma *conway-point-past-second-contact*:
  fixes X Y :: $'a$::*real-normed-vector*
  assumes $X \neq Y$
  shows *conway-point-past-second* X Y q =
    side-contact-point X Y p q + (((p + q + *dist* X Y) / *2*) / *dist* X Y) $*_R$ (Y −
X)
proof −
  let $?L$ = *dist* X Y
  let $?s$ = (p + q + $?L$) / *2*
  have *Lpos*: *0* < $?L$
    using *assms* **by** *simp*
  have *scalar*: (($?s$ − p) / $?L$ + $?s$ / $?L$) = *1* + q / $?L$
    using *Lpos* **by** (*simp add*: *field-simps*)
  have *side-contact-point* X Y p q + ($?s$ / $?L$) $*_R$ (Y − X) =
      X + (($?s$ − p) / $?L$) $*_R$ (Y − X) + ($?s$ / $?L$) $*_R$ (Y − X)
    unfolding *side-contact-point-def Let-def* **by** *simp*
  also have ... = X + ((($?s$ − p) / $?L$ + $?s$ / $?L$) $*_R$ (Y − X))
    by (*simp add*: *scaleR-left-distrib* [*symmetric*])
  also have ... = X + (*1* + q / $?L$) $*_R$ (Y − X)
    using *scalar* **by** *simp*
  also have ... = Y + (q / $?L$) $*_R$ (Y − X)
    by (*simp add*: *scaleR-left-distrib*)
  also have ... = *conway-point-past-second* X Y q
    unfolding *conway-point-past-second-def* **..**
  finally show *?thesis*
    by (*rule sym*)
qed

lemma *dist-side-contact-first*:
  fixes X Y :: $'a$::*real-normed-vector*
  assumes $X \neq Y$ $0 \leq p$ $0 \leq q$
  shows *dist* (*side-contact-point* X Y p q) (*conway-point-past-first* X Y p) =
    (p + q + *dist* X Y) / *2*
proof −
  let $?s$ = (p + q + *dist* X Y) / *2*
  have *Lpos*: *0* < *dist* X Y
    using *assms* **by** *simp*
  have *spos*: *0* ≤ $?s$
    using *assms Lpos* **by** *simp*

```
  have point-eq:
    conway-point-past-first X Y p =
      side-contact-point X Y p q − (?s / dist X Y) *R (Y − X)
    by (rule conway-point-past-first-contact[OF assms(1)])
  have diff-eq:
    side-contact-point X Y p q − conway-point-past-first X Y p =
      (?s / dist X Y) *R (Y − X)
    using point-eq by simp
  have dist (side-contact-point X Y p q) (conway-point-past-first X Y p) =
      norm ((?s / dist X Y) *R (Y − X))
    by (simp add: dist-norm diff-eq)
  also have ... = abs (?s / dist X Y) * norm (Y − X)
    by simp
  also have ... = (?s / dist X Y) * dist X Y
    using Lpos spos by (simp add: dist-norm norm-minus-commute)
  also have ... = ?s
    using Lpos by simp
  finally show ?thesis .
qed

lemma dist-side-contact-second:
  fixes X Y :: 'a::real-normed-vector
  assumes X ≠ Y 0 ≤ p 0 ≤ q
  shows dist (side-contact-point X Y p q) (conway-point-past-second X Y q) =
    (p + q + dist X Y) / 2
proof −
  let ?s = (p + q + dist X Y) / 2
  have Lpos: 0 < dist X Y
    using assms by simp
  have spos: 0 ≤ ?s
    using assms Lpos by simp
  have point-eq:
    conway-point-past-second X Y q =
      side-contact-point X Y p q + (?s / dist X Y) *R (Y − X)
    by (rule conway-point-past-second-contact[OF assms(1)])
  have diff-eq:
    side-contact-point X Y p q − conway-point-past-second X Y q =
      − (?s / dist X Y) *R (Y − X)
    using point-eq by simp
  have dist (side-contact-point X Y p q) (conway-point-past-second X Y q) =
      norm ((?s / dist X Y) *R (Y − X))
    by (simp add: dist-norm diff-eq)
  also have ... = abs (?s / dist X Y) * norm (Y − X)
    by simp
  also have ... = (?s / dist X Y) * dist X Y
    using Lpos spos by (simp add: dist-norm norm-minus-commute)
  also have ... = ?s
    using Lpos by simp
  finally show ?thesis .
```

qed

lemma *dist-side-contact-first′*:
  fixes *X Y* :: *′a*::*real-normed-vector*
  assumes $X \neq Y$ $0 \leq p$ $0 \leq q$
  shows *dist* (*conway-point-past-first X Y p*) (*side-contact-point X Y p q*) =
    (*p* + *q* + *dist X Y*) / *2*
  using *dist-side-contact-first*[*OF assms*] **by** (*simp add*: *dist-commute*)

lemma *dist-side-contact-second′*:
  fixes *X Y* :: *′a*::*real-normed-vector*
  assumes $X \neq Y$ $0 \leq p$ $0 \leq q$
  shows *dist* (*conway-point-past-second X Y q*) (*side-contact-point X Y p q*) =
    (*p* + *q* + *dist X Y*) / *2*
  using *dist-side-contact-second*[*OF assms*] **by** (*simp add*: *dist-commute*)

lemma *orthogonal-conway-point-past-first-contact*:
  fixes *X Y v* :: *′a*::*real-inner*
  assumes $X \neq Y$ *orthogonal v* (*Y* − *X*)
  shows *orthogonal v* (*conway-point-past-first X Y p* − *side-contact-point X Y p q*)
proof −
  let *?s* = (*p* + *q* + *dist X Y*) / *2*
  have *point-eq*:
    conway-point-past-first X Y p − *side-contact-point X Y p q* =
      − (*?s* / *dist X Y*) $*_R$ (*Y* − *X*)
    using *conway-point-past-first-contact*[*OF assms*(*1*), **where** *p*=*p* **and** *q*=*q*] **by** *simp*
  show *?thesis*
    using *assms*(*2*) **unfolding** *point-eq orthogonal-def* **by** *simp*
qed

lemma *orthogonal-conway-point-past-second-contact*:
  fixes *X Y v* :: *′a*::*real-inner*
  assumes $X \neq Y$ *orthogonal v* (*Y* − *X*)
  shows *orthogonal v* (*conway-point-past-second X Y q* − *side-contact-point X Y p q*)
proof −
  let *?s* = (*p* + *q* + *dist X Y*) / *2*
  have *point-eq*:
    conway-point-past-second X Y q − *side-contact-point X Y p q* =
      (*?s* / *dist X Y*) $*_R$ (*Y* − *X*)
    using *conway-point-past-second-contact*[*OF assms*(*1*), **where** *p*=*p* **and** *q*=*q*] **by** *simp*
  show *?thesis*
    using *assms*(*2*) **unfolding** *point-eq orthogonal-def* **by** *simp*
qed

lemma *dist-from-orthogonal-foot*:
  fixes *I T P* :: *′a*::*real-inner*

assumes *orth*: *orthogonal* $(I - T)$ $(P - T)$
  and *IT*: *dist* I $T = r$
  and *TP*: *dist* T $P = s$
  and *r-nonneg*: $0 \leq r$
  and *s-nonneg*: $0 \leq s$
shows *dist* I $P =$ *sqrt* $(r^2 + s^2)$
proof $-$
  have *inner-flip*: $(I - T) \cdot (T - P) = - ((I - T) \cdot (P - T))$
    by (*simp add*: *inner-diff-right*)
  have *orth'*: *orthogonal* $(I - T)$ $(T - P)$
    using *orth inner-flip* **by** (*simp add*: *orthogonal-def*)
  have *Pythagoras*:
    (*norm* $((I - T) + (T - P)))^2 =$
      (*norm* $(I - T))^2 +$ (*norm* $(T - P))^2$
    by (*rule norm-add-Pythagorean*[*OF orth'*])
  have *square-dist*:
    (*dist* I $P)^2 = r^2 + s^2$
    using *Pythagoras IT TP*
    by (*simp add*: *dist-norm norm-minus-commute*)
  have *dist* I $P =$ *sqrt* ((*dist* I $P)^2$)
    by *simp*
  also have ... $=$ *sqrt* $(r^2 + s^2)$
    using *square-dist* **by** *simp*
  finally show *?thesis* **.**
qed

lemma *conway-points-have-declared-lengths*:
  assumes $A \neq B$ $A \neq C$ $B \neq C$
  shows
    dist A (*conway-A-c* A B C) $=$ *dist* B C
    dist B (*conway-B-c* A B C) $=$ *dist* C A
    dist A (*conway-A-b* A B C) $=$ *dist* B C
    dist C (*conway-C-b* A B C) $=$ *dist* A B
    dist B (*conway-B-a* A B C) $=$ *dist* C A
    dist C (*conway-C-a* A B C) $=$ *dist* A B
  using *assms*
  unfolding *conway-A-c-def conway-B-c-def conway-A-b-def conway-C-b-def*
    conway-B-a-def conway-C-a-def
  by (*simp-all add*: *dist-conway-point-past-first dist-conway-point-past-second*)

lemma *conway-points-about-contacts*:
  assumes $A \neq B$ $A \neq C$ $B \neq C$
  defines $s \equiv$ *triangle-semiperimeter* A B C
  shows
    dist (*contact-AB* A B C) (*conway-A-c* A B C) $= s$
    dist (*contact-AB* A B C) (*conway-B-c* A B C) $= s$
    dist (*contact-AC* A B C) (*conway-A-b* A B C) $= s$
    dist (*contact-AC* A B C) (*conway-C-b* A B C) $= s$
    dist (*contact-BC* A B C) (*conway-B-a* A B C) $= s$

dist (*contact-BC A B C*) (*conway-C-a A B C*) = *s*
using *assms*
unfolding *s-def triangle-semiperimeter-def*
contact-AB-def contact-AC-def contact-BC-def
conway-A-c-def conway-B-c-def conway-A-b-def conway-C-b-def
conway-B-a-def conway-C-a-def
by (*simp-all add*: *dist-side-contact-first dist-side-contact-second*
dist-side-contact-first′ dist-side-contact-second′ dist-commute)

lemma *conway-points-on-circle-from-contact-feet*:
fixes *A B C I* :: *real* ^ *2*
assumes *sides*: $A \neq B$ $A \neq C$ $B \neq C$
and *r-nonneg*: $0 \leq r$
and *foot-AB*: *dist I* (*contact-AB A B C*) = *r*
orthogonal (*I* − *contact-AB A B C*) (*B* − *A*)
and *foot-AC*: *dist I* (*contact-AC A B C*) = *r*
orthogonal (*I* − *contact-AC A B C*) (*C* − *A*)
and *foot-BC*: *dist I* (*contact-BC A B C*) = *r*
orthogonal (*I* − *contact-BC A B C*) (*C* − *B*)
defines *s* ≡ *triangle-semiperimeter A B C*
shows
dist I (*conway-A-c A B C*) = *sqrt* (r^2 + s^2)
dist I (*conway-B-c A B C*) = *sqrt* (r^2 + s^2)
dist I (*conway-A-b A B C*) = *sqrt* (r^2 + s^2)
dist I (*conway-C-b A B C*) = *sqrt* (r^2 + s^2)
dist I (*conway-B-a A B C*) = *sqrt* (r^2 + s^2)
dist I (*conway-C-a A B C*) = *sqrt* (r^2 + s^2)
proof −
have *s-nonneg*: $0 \leq s$
unfolding *s-def triangle-semiperimeter-def* **by** *simp*
have *contact-dists*:
dist (*contact-AB A B C*) (*conway-A-c A B C*) = *s*
dist (*contact-AB A B C*) (*conway-B-c A B C*) = *s*
dist (*contact-AC A B C*) (*conway-A-b A B C*) = *s*
dist (*contact-AC A B C*) (*conway-C-b A B C*) = *s*
dist (*contact-BC A B C*) (*conway-B-a A B C*) = *s*
dist (*contact-BC A B C*) (*conway-C-a A B C*) = *s*
using *conway-points-about-contacts*[*OF sides*, *folded s-def*] **.**
have *orth-Ac*:
orthogonal (*I* − *contact-AB A B C*) (*conway-A-c A B C* − *contact-AB A B C*)
using *foot-AB*(*2*)
unfolding *conway-A-c-def contact-AB-def*
by (*rule orthogonal-conway-point-past-first-contact*[*OF sides*(*1*)])
have *orth-Bc*:
orthogonal (*I* − *contact-AB A B C*) (*conway-B-c A B C* − *contact-AB A B C*)
using *foot-AB*(*2*)
unfolding *conway-B-c-def contact-AB-def*

by (*rule orthogonal-conway-point-past-second-contact*[*OF sides*(*1*)])
have *orth-Ab*:
orthogonal (*I* − *contact-AC A B C*) (*conway-A-b A B C* − *contact-AC A B C*)
using *foot-AC*(*2*)
unfolding *conway-A-b-def contact-AC-def*
by (*rule orthogonal-conway-point-past-first-contact*[*OF sides*(*2*)])
have *orth-Cb*:
orthogonal (*I* − *contact-AC A B C*) (*conway-C-b A B C* − *contact-AC A B C*)
using *foot-AC*(*2*)
unfolding *conway-C-b-def contact-AC-def*
by (*rule orthogonal-conway-point-past-second-contact*[*OF sides*(*2*)])
have *orth-Ba*:
orthogonal (*I* − *contact-BC A B C*) (*conway-B-a A B C* − *contact-BC A B C*)
using *foot-BC*(*2*)
unfolding *conway-B-a-def contact-BC-def*
by (*rule orthogonal-conway-point-past-first-contact*[*OF sides*(*3*)])
have *orth-Ca*:
orthogonal (*I* − *contact-BC A B C*) (*conway-C-a A B C* − *contact-BC A B C*)
using *foot-BC*(*2*)
unfolding *conway-C-a-def contact-BC-def*
by (*rule orthogonal-conway-point-past-second-contact*[*OF sides*(*3*)])
show *dist I* (*conway-A-c A B C*) = *sqrt* (r^2 + s^2)
by (*rule dist-from-orthogonal-foot*[*OF orth-Ac foot-AB*(*1*) *contact-dists*(*1*) *r-nonneg s-nonneg*])
show *dist I* (*conway-B-c A B C*) = *sqrt* (r^2 + s^2)
by (*rule dist-from-orthogonal-foot*[*OF orth-Bc foot-AB*(*1*) *contact-dists*(*2*) *r-nonneg s-nonneg*])
show *dist I* (*conway-A-b A B C*) = *sqrt* (r^2 + s^2)
by (*rule dist-from-orthogonal-foot*[*OF orth-Ab foot-AC*(*1*) *contact-dists*(*3*) *r-nonneg s-nonneg*])
show *dist I* (*conway-C-b A B C*) = *sqrt* (r^2 + s^2)
by (*rule dist-from-orthogonal-foot*[*OF orth-Cb foot-AC*(*1*) *contact-dists*(*4*) *r-nonneg s-nonneg*])
show *dist I* (*conway-B-a A B C*) = *sqrt* (r^2 + s^2)
by (*rule dist-from-orthogonal-foot*[*OF orth-Ba foot-BC*(*1*) *contact-dists*(*5*) *r-nonneg s-nonneg*])
show *dist I* (*conway-C-a A B C*) = *sqrt* (r^2 + s^2)
by (*rule dist-from-orthogonal-foot*[*OF orth-Ca foot-BC*(*1*) *contact-dists*(*6*) *r-nonneg s-nonneg*])
qed

lemma *conway-points-on-circle-from-noncollinear-contact-feet*:
fixes *A B C I* :: *real* ^ *2*
assumes *triangle*: ¬ *collinear* {*A*, *B*, *C*}
and *r-nonneg*: *0* ≤ *r*
and *foot-AB*: *dist I* (*contact-AB A B C*) = *r*
orthogonal (*I* − *contact-AB A B C*) (*B* − *A*)

and *foot-AC*: *dist I* (*contact-AC A B C*) = *r*
orthogonal (*I* − *contact-AC A B C*) (*C* − *A*)
and *foot-BC*: *dist I* (*contact-BC A B C*) = *r*
orthogonal (*I* − *contact-BC A B C*) (*C* − *B*)
defines *s* ≡ *triangle-semiperimeter A B C*
shows
dist I (*conway-A-c A B C*) = *sqrt* (r^2 + s^2)
dist I (*conway-B-c A B C*) = *sqrt* (r^2 + s^2)
dist I (*conway-A-b A B C*) = *sqrt* (r^2 + s^2)
dist I (*conway-C-b A B C*) = *sqrt* (r^2 + s^2)
dist I (*conway-B-a A B C*) = *sqrt* (r^2 + s^2)
dist I (*conway-C-a A B C*) = *sqrt* (r^2 + s^2)
proof −
have *sides*: *A* ≠ *B A* ≠ *C B* ≠ *C*
using *noncollinear-triangle-distinct*[*OF triangle*] **by** *blast+*
show *dist I* (*conway-A-c A B C*) = *sqrt* (r^2 + s^2)
and *dist I* (*conway-B-c A B C*) = *sqrt* (r^2 + s^2)
and *dist I* (*conway-A-b A B C*) = *sqrt* (r^2 + s^2)
and *dist I* (*conway-C-b A B C*) = *sqrt* (r^2 + s^2)
and *dist I* (*conway-B-a A B C*) = *sqrt* (r^2 + s^2)
and *dist I* (*conway-C-a A B C*) = *sqrt* (r^2 + s^2)
using *conway-points-on-circle-from-contact-feet*
[*OF sides r-nonneg foot-AB foot-AC foot-BC*, *folded s-def*] **.**
qed

theorem *conway-circle-theorem*:
fixes *A B C* :: *real* ^ *2*
assumes *triangle*: ¬ *collinear* {*A*, *B*, *C*}
defines *I* ≡ *triangle-incenter A B C*
defines *r* ≡ *triangle-inradius A B C*
defines *s* ≡ *triangle-semiperimeter A B C*
defines *A-c* ≡ *conway-A-c A B C*
defines *B-c* ≡ *conway-B-c A B C*
defines *A-b* ≡ *conway-A-b A B C*
defines *C-b* ≡ *conway-C-b A B C*
defines *B-a* ≡ *conway-B-a A B C*
defines *C-a* ≡ *conway-C-a A B C*
shows
dist I A-c = *sqrt* (r^2 + s^2)
dist I B-c = *sqrt* (r^2 + s^2)
dist I A-b = *sqrt* (r^2 + s^2)
dist I C-b = *sqrt* (r^2 + s^2)
dist I B-a = *sqrt* (r^2 + s^2)
dist I C-a = *sqrt* (r^2 + s^2)
proof −
have *foot-dists*:
dist I (*contact-AB A B C*) = *r*
dist I (*contact-AC A B C*) = *r*
dist I (*contact-BC A B C*) = *r*

using *triangle-incenter-contacts-dist*[*OF triangle*]
unfolding *I-def r-def* **by** *simp-all*
have *foot-orthogonal*:
orthogonal $(I - contact\text{-}AB\ A\ B\ C)$ $(B - A)$
orthogonal $(I - contact\text{-}AC\ A\ B\ C)$ $(C - A)$
orthogonal $(I - contact\text{-}BC\ A\ B\ C)$ $(C - B)$
using *triangle-incenter-contacts-orthogonal*[*OF triangle*]
unfolding *I-def* **by** *simp-all*
have *r-nonneg*: $0 \le r$
unfolding *r-def* **by** *simp*
have *circle*:
dist I (*conway-A-c A B C*) = *sqrt* $(r^2 + s^2)$
dist I (*conway-B-c A B C*) = *sqrt* $(r^2 + s^2)$
dist I (*conway-A-b A B C*) = *sqrt* $(r^2 + s^2)$
dist I (*conway-C-b A B C*) = *sqrt* $(r^2 + s^2)$
dist I (*conway-B-a A B C*) = *sqrt* $(r^2 + s^2)$
dist I (*conway-C-a A B C*) = *sqrt* $(r^2 + s^2)$
using *conway-points-on-circle-from-noncollinear-contact-feet*
[*OF triangle r-nonneg foot-dists*(*1*) *foot-orthogonal*(*1*)
foot-dists(*2*) *foot-orthogonal*(*2*) *foot-dists*(*3*) *foot-orthogonal*(*3*),
folded s-def] **.**
show *dist I A-c* = *sqrt* $(r^2 + s^2)$
unfolding *A-c-def* **by** (*rule circle*(*1*))
show *dist I B-c* = *sqrt* $(r^2 + s^2)$
unfolding *B-c-def* **by** (*rule circle*(*2*))
show *dist I A-b* = *sqrt* $(r^2 + s^2)$
unfolding *A-b-def* **by** (*rule circle*(*3*))
show *dist I C-b* = *sqrt* $(r^2 + s^2)$
unfolding *C-b-def* **by** (*rule circle*(*4*))
show *dist I B-a* = *sqrt* $(r^2 + s^2)$
unfolding *B-a-def* **by** (*rule circle*(*5*))
show *dist I C-a* = *sqrt* $(r^2 + s^2)$
unfolding *C-a-def* **by** (*rule circle*(*6*))
qed

4.1 Wrapper formulations

The headline theorem above proves six separate distance equations. For downstream use we package the result in two equivalent forms: as a subset of an explicit *sphere* centred at the incenter, and as a single bounded-universal statement over the explicit six-point set. We also expose the construction correctness as a separate theorem: each Conway point lies on the intended side line and is at the declared distance from the adjacent vertex.

lemma *conway-point-past-first-on-line*:
fixes *X Y* :: *'a::real-normed-vector*
shows *conway-point-past-first X Y d* $= X + (- (d\ /\ dist\ X\ Y)) *_R (Y - X)$
unfolding *conway-point-past-first-def*
by (*simp add*: *scaleR-minus-right algebra-simps*)

```
lemma conway-point-past-second-on-line:
  fixes X Y :: 'a::real-normed-vector
  shows conway-point-past-second X Y d
          = X + (1 + d / dist X Y) *R (Y − X)
  unfolding conway-point-past-second-def
  by (simp add: scaleR-left-distrib)

theorem conway-construction-correct:
  fixes A B C :: real ^ 2
  assumes triangle: ¬ collinear {A, B, C}
  shows
    — The six Conway points lie on the three side lines of the triangle.
    ∃ t::real. conway-A-c A B C = A + t *R (B − A)
    ∃ t::real. conway-B-c A B C = A + t *R (B − A)
    ∃ t::real. conway-A-b A B C = A + t *R (C − A)
    ∃ t::real. conway-C-b A B C = A + t *R (C − A)
    ∃ t::real. conway-B-a A B C = B + t *R (C − B)
    ∃ t::real. conway-C-a A B C = B + t *R (C − B)
    — Their distances from the adjacent vertex equal the opposite side length.
    dist A (conway-A-c A B C) = dist B C
    dist B (conway-B-c A B C) = dist C A
    dist A (conway-A-b A B C) = dist B C
    dist C (conway-C-b A B C) = dist A B
    dist B (conway-B-a A B C) = dist C A
    dist C (conway-C-a A B C) = dist A B
proof −
  have sides: A ≠ B A ≠ C B ≠ C
    using noncollinear-triangle-distinct[OF triangle] by blast+
  show
    ∃ t::real. conway-A-c A B C = A + t *R (B − A)
    unfolding conway-A-c-def
    using conway-point-past-first-on-line[of A B dist B C] by blast
  show
    ∃ t::real. conway-B-c A B C = A + t *R (B − A)
    unfolding conway-B-c-def
    using conway-point-past-second-on-line[of A B dist C A] by blast
  show
    ∃ t::real. conway-A-b A B C = A + t *R (C − A)
    unfolding conway-A-b-def
    using conway-point-past-first-on-line[of A C dist B C] by blast
  show
    ∃ t::real. conway-C-b A B C = A + t *R (C − A)
    unfolding conway-C-b-def
    using conway-point-past-second-on-line[of A C dist A B] by blast
  show
    ∃ t::real. conway-B-a A B C = B + t *R (C − B)
    unfolding conway-B-a-def
    using conway-point-past-first-on-line[of B C dist C A] by blast
```

show
$\exists\, t{::}real.\ conway\text{-}C\text{-}a\ A\ B\ C = B + t *_R (C - B)$
unfolding *conway-C-a-def*
using *conway-point-past-second-on-line*[*of B C dist A B*] **by** *blast*
show
$dist\ A\ (conway\text{-}A\text{-}c\ A\ B\ C) = dist\ B\ C$
$dist\ B\ (conway\text{-}B\text{-}c\ A\ B\ C) = dist\ C\ A$
$dist\ A\ (conway\text{-}A\text{-}b\ A\ B\ C) = dist\ B\ C$
$dist\ C\ (conway\text{-}C\text{-}b\ A\ B\ C) = dist\ A\ B$
$dist\ B\ (conway\text{-}B\text{-}a\ A\ B\ C) = dist\ C\ A$
$dist\ C\ (conway\text{-}C\text{-}a\ A\ B\ C) = dist\ A\ B$
using *conway-points-have-declared-lengths*[*OF sides*] **by** *blast+*
qed

definition *conway-points* :: $real \hat{}\ 2 \Rightarrow real \hat{}\ 2 \Rightarrow real \hat{}\ 2 \Rightarrow (real \hat{}\ 2)\ set$
where *conway-points A B C* =
$\{conway\text{-}A\text{-}c\ A\ B\ C,\ conway\text{-}B\text{-}c\ A\ B\ C,\ conway\text{-}A\text{-}b\ A\ B\ C,$
$conway\text{-}C\text{-}b\ A\ B\ C,\ conway\text{-}B\text{-}a\ A\ B\ C,\ conway\text{-}C\text{-}a\ A\ B\ C\}$

corollary *conway-circle-theorem-sphere*:
fixes $A\ B\ C :: real \hat{}\ 2$
assumes *triangle*: $\neg\ collinear\ \{A, B, C\}$
shows
conway-points A B C
$\subseteq sphere\ (triangle\text{-}incenter\ A\ B\ C)$
$(sqrt\ ((triangle\text{-}inradius\ A\ B\ C)^2$
$+ (triangle\text{-}semiperimeter\ A\ B\ C)^2))$
using *conway-circle-theorem*[*OF triangle*]
unfolding *conway-points-def sphere-def* **by** *auto*

corollary *conway-circle-theorem-set*:
fixes $A\ B\ C :: real \hat{}\ 2$
assumes *triangle*: $\neg\ collinear\ \{A, B, C\}$
shows
$\forall P \in conway\text{-}points\ A\ B\ C.$
$dist\ (triangle\text{-}incenter\ A\ B\ C)\ P$
$= sqrt\ ((triangle\text{-}inradius\ A\ B\ C)^2$
$+ (triangle\text{-}semiperimeter\ A\ B\ C)^2)$
using *conway-circle-theorem*[*OF triangle*]
unfolding *conway-points-def* **by** *auto*

end

References

[1] F. J. G. Capitán. A generalization of the Conway circle. *Forum Geometricorum*, 13:191–195, 2013. Available at https://forumgeom.fau.edu/FG2013volume13/FG201320.pdf.

[2] L. S. Evans. A conic through six triangle centers. *Forum Geometricorum*, 2:89–92, 2002. Available at https://forumgeom.fau.edu/FG2002volume2/FG200211.pdf.

[3] D. Grinberg and E. W. Weisstein. Conway Circle. MathWorld–A Wolfram Web Resource. Available at https://mathworld.wolfram.com/ConwayCircle.html; accessed 2026-05-13.

[4] R. Honsberger. *Episodes in Nineteenth and Twentieth Century Euclidean Geometry*, volume 37 of *New Mathematical Library*. Mathematical Association of America, 1995.