Proving the Impossibility of Trisecting an Angle and Doubling the Cube

Ralph Romanos and Lawrence Paulson

August 16, 2018

Abstract

Squaring the circle, doubling the cube and trisecting an angle, using a compass and straightedge alone, are classic unsolved problems first posed by the ancient Greeks. All three problems were proved to be impossible in the 19th century. The following document presents the proof of the impossibility of solving the latter two problems using Isabelle/HOL, following a proof by Carrega [Car81]. The proof uses elementary methods: no Galois theory or field extensions. The set of points constructible using a compass and straightedge is defined inductively. Radical expressions, which involve only square roots and arithmetic of rational numbers, are defined, and we find that all constructive points have radical coordinates. Finally, doubling the cube and trisecting certain angles requires solving certain cubic equations that can be proved to have no rational roots. The Isabelle proofs require a great many detailed calculations.

Contents

1 Proving the impossibility of trisecting an angle and doubling the cube 2

2 Formal Proof 2

2.1 Definition of the set of Points 2
2.2 Subtraction 2
2.3 Metric Space 4
2.4 Geometric Definitions 5
2.5 Reals definable with square roots 6
2.6 Introduction of the datatype expr which represents radical expressions 6
2.7 Important properties of the roots of a cubic equation 14
2.8 Important properties of radicals 18
2.9 Important properties of geometrical points which coordinates are radicals 24
2.10 Definition of the set of constructible points .......................... 27
2.11 An important property about constructible points: their co-
oordinates are radicals .............................................. 28
2.12 Proving the impossibility of duplicating the cube ................. 28
2.13 Proving the impossibility of trisecting an angle ...................... 29

1 Proving the impossibility of trisecting an angle
and doubling the cube

theory Impossible-Geometry
imports Complex-Main
begin

2 Formal Proof

2.1 Definition of the set of Points

datatype point = Point real real

definition points-def:
points = {M. ∃ x ∈ R. ∃ y ∈ R. (M = Point x y)}

primrec abscissa :: point => real
where abscissa: abscissa (Point x y) = x

primrec ordinate :: point => real
where ordinate: ordinate (Point x y) = y

lemma point-surj [simp]:
Point (abscissa M) (ordinate M) = M
by (induct M) simp

lemma point-eqI [intro?):
[abscissa M = abscissa N; ordinate M = ordinate N] => M = N
by (induct M, induct N) simp

lemma point-eq-iff:
M = N <-> abscissa M = abscissa N ∧ ordinate M = ordinate N
by (induct M, induct N) simp

2.2 Subtraction

Datatype point has a structure of abelian group

instantiation point :: ab-group-add
begin

definition point-zero-def:
\[0 = Point \ 0 \ 0\]

**definition** point-one-def:
\[point-one = Point \ 1 \ 0\]

**definition** point-add-def:
\[A + B = Point \ (\text{abscissa} \ A + \text{abscissa} \ B) \ (\text{ordinate} \ A + \text{ordinate} \ B)\]

**definition** point-minus-def:
\[-A = Point \ (-\ \text{abscissa} \ A) \ (-\ \text{ordinate} \ A)\]

**definition** point-diff-def:
\[A - (B::point) = A + (-B)\]

**lemma** Point-eq-0 [simp]:
\[Point \ xA \ yA = 0 \leftrightarrow (xA = 0 \wedge yA = 0)\]
by (simp add: point-zero-def)

**lemma** point-abscissa-zero [simp]:
\[\text{abscissa} \ 0 = 0\]
by (simp add: point-zero-def)

**lemma** point-ordinate-zero [simp]:
\[\text{ordinate} \ 0 = 0\]
by (simp add: point-zero-def)

**lemma** point-add [simp]:
\[Point \ xA \ yA + Point \ xB \ yB = Point \ (xA + xB) \ (yA + yB)\]
by (simp add: point-add-def)

**lemma** point-abscissa-add [simp]:
\[\text{abscissa} \ (A + B) = \text{abscissa} \ A + \text{abscissa} \ B\]
by (simp add: point-add-def)

**lemma** point-ordinate-add [simp]:
\[\text{ordinate} \ (A + B) = \text{ordinate} \ A + \text{ordinate} \ B\]
by (simp add: point-add-def)

**lemma** point-minus [simp]:
\[-(Point \ xA \ yA) = Point \ (-xA) \ (-yA)\]
by (simp add: point-minus-def)

**lemma** point-abscissa-minus [simp]:
\[\text{abscissa} \ (-A) = - \text{abscissa} \ (A)\]
by (simp add: point-minus-def)

**lemma** point-ordinate-minus [simp]:
\[\text{ordinate} \ (-A) = - \text{ordinate} \ (A)\]
by (simp add: point-minus-def)
lemma point-diff [simp]:
Point xA yA − Point xB yB = Point (xA − xB) (yA − yB)
by (simp add: point-diff-def)

lemma point-abscissa-diff [simp]:
abscissa (A − B) = abscissa (A) − abscissa (B)
by (simp add: point-diff-def)

lemma point-ordinate-diff [simp]:
ordinate (A − B) = ordinate (A) − ordinate (B)
by (simp add: point-diff-def)

instance
by intro-classes (simp-all add: point-add-def point-diff-def)
end

2.3 Metric Space

We can also define a distance, hence point is also a metric space

instantiation point :: metric-space
begin

definition point-dist-def:
dist A B = sqrt ((abscissa (A − B)) ^ 2 + (ordinate (A − B)) ^ 2)

definition
(uniformity :: (point × point) filter) = (INF e:{0 <..}. principal {(x, y). dist x y < e})

definition
open (S :: point set) = (∀ x∈S. ∀ F (x', y) in uniformity. x' = x −→ y ∈ S)

lemma point-dist [simp]:
dist (Point xA yA) (Point xB yB) = sqrt ((xA − xB) ^ 2 + (yA − yB) ^ 2)
unfolding point-dist-def
by simp

lemma real-sqrt-diff-squares-triangle-ineq:
fixes a b c d :: real
shows sqrt ((a − c) ^ 2 + (b − d) ^ 2) ≤ sqrt (a ^ 2 + b ^ 2) + sqrt (c ^ 2 + d ^ 2)
proof –
have sqrt ((a − c) ^ 2 + (b − d) ^ 2) ≤ sqrt (a ^ 2 + b ^ 2) + sqrt ((−c) ^ 2 + (−d) ^ 2)
by (metis diff-conv-add-uminus real-sqrt-sum-squares-triangle-ineq)
also have ... = sqrt (a ^ 2 + b ^ 2) + sqrt (c ^ 2 + d ^ 2)
by simp
finally show ?thesis .

4
qed

instance

proof
  fix A B C :: point and S :: point set
  show (dist A B = 0) = (A = B)
    by (induct A, induct B) (simp add: point-dist-def)
  show (dist A B) ≤ (dist A C) + (dist B C)
    proof
      have sqrt ((abscissa (A - B)) ^ 2 + (ordinate (A - B)) ^ 2) ≤
        sqrt ((abscissa (A - C)) ^ 2 + (ordinate (A - C)) ^ 2) +
        sqrt ((abscissa (B - C)) ^ 2 + (ordinate (B - C)) ^ 2)
        using real-sqrt-diff-squares-triangle-ineq
        [of abscissa (A - abscissa (C) abscissa (B) - abscissa (C) ordinate (A) - ordinate (C) ordinate (B) - ordinate (C))]
      thus ?thesis
        by (simp only: point-dist-def (simp add: algebra-simps)
      qed
    qed
qed (rule uniformity-point-def open-point-def)

end

2.4 Geometric Definitions

These geometric definitions will later be used to define constructible points

The distance between two points is defined with the distance of the metric space point

definition distance-def:
  distance A B = dist A B

parallel A B C D is true if the lines AB and CD are parallel. If not it is false.

definition parallel-def:
  parallel A B C D = ((abscissa A - abscissa B) * (ordinate C - ordinate D) =
    (ordinate A - ordinate B) * (abscissa C - abscissa D))

Three points A B C are collinear if and only if the lines AB and AC are parallel

definition collinear-def:
  collinear A B C = parallel A B A C

The point M is the intersection of two lines AB and CD if and only if
the points A, M and B are collinear and the points C, M and D are also collinear

definition is-intersection-def:
  is-intersection M A B C D = (collinear A M B ∧ collinear C M D)
2.5 Reals definable with square roots

The inductive set $\text{radical-sqrt}$ defines the reals that can be defined with square roots. If $x$ is in the following set, then it depends only upon rational expressions and square roots. For example, suppose $x$ is of the form:

$$x = (\sqrt{a + \sqrt{b}} + \sqrt{c + \sqrt{d + e + f}})/\sqrt{g}, \quad \text{where} \quad a, b, c, d, e, f, \text{and} \ g \ \text{are rationals}. \quad \text{Then} \quad x \ \text{is in radical-sqrt because it is only defined with rationals and square roots of radicals}.$$

**inductive-set** $\text{radical-sqrt} :: \text{real set}$

where

- $x \in \mathbb{Q} \implies x \in \text{radical-sqrt}$
- $x \in \text{radical-sqrt}, y \in \text{radical-sqrt} \implies x \neq 0 \implies 1/x \in \text{radical-sqrt}$
- $x \in \text{radical-sqrt}, y \in \text{radical-sqrt} \implies x + y \in \text{radical-sqrt}$
- $x \in \text{radical-sqrt}, y \in \text{radical-sqrt} \implies x \cdot y \in \text{radical-sqrt}$
- $x \in \text{radical-sqrt}, x \geq 0 \implies \sqrt{x} \in \text{radical-sqrt}$

Here, we list some rules that will be used to prove that a given real is in radical-sqrt.

Given two reals in radical-sqrt $x$ and $y$, the subtraction $x - y$ is also in radical-sqrt.

**lemma** radical-sqrt-rule-subtraction:

- $x \in \text{radical-sqrt} \implies y \in \text{radical-sqrt} \implies x - y \in \text{radical-sqrt}$

by (metis diff-conv-add-uminus radical-sqrt.

Given two reals in radical-sqrt $x$ and $y$, and $y \neq 0$, the division $x/y$ is also in radical-sqrt.

**lemma** radical-sqrt-rule-division:

- $x \in \text{radical-sqrt} \implies y \in \text{radical-sqrt} \implies y \neq 0 \implies x/y \in \text{radical-sqrt}$

by (metis divide-real-def radical-sqrt.

Given a positive real $x$ in radical-sqrt, its square $x^2$ is also in radical-sqrt.

**lemma** radical-sqrt-rule-power2:

- $x \in \text{radical-sqrt} \implies x \geq 0 \implies x^2 \in \text{radical-sqrt}$

by (metis power2-eq-square radical-sqrt.

Given a positive real $x$ in radical-sqrt, its cube $x^3$ is also in radical-sqrt.

**lemma** radical-sqrt-rule-power3:

- $x \in \text{radical-sqrt} \implies x \geq 0 \implies x^3 \in \text{radical-sqrt}$

by (metis power3-eq-cube radical-sqrt.

2.6 Introduction of the datatype expr which represents radical expressions

An expression expr is either a rational constant: Const or the negation of an expression or the inverse of an expression or the addition of two expressions
or the multiplication of two expressions or the square root of an expression.

**datatype** expr = Const rat | Negation expr | Inverse expr | Addition expr expr | Multiplication expr expr | Sqrt expr

The function *translation* translates a given expression into its equivalent real.

```haskell
fun translation :: expr => real ((2\frac{-1}{2}))
where
  translation (Const x) = of-rat x
  translation (Negation e) = − translation e
  translation (Inverse e) = \frac{1::real}{translation e}
  translation (Addition e1 e2) = translation e1 + translation e2
  translation (Multiplication e1 e2) = translation e1 * translation e2
  translation (Sqrt e) = (if translation e < 0 then 0 else sqrt (translation e))
```

Define the set of all the radicals of a given expression. For example, suppose `expr` is of the form:

```
expr = Addition (Sqrt (Addition (Const a) Sqrt (Const b))) (Sqrt (Addition (Const c) (Sqrt (Sqrt (Const d)))))
```

where `a`, `b`, `c` and `d` are rationals. This can be translated as follows: \[\|expr\| = \sqrt{a + \sqrt{b + \sqrt{c + \sqrt{d}}}}.\] Moreover, the set `radicals` of this expression is:

```
\{\|Addition (Const a) (Sqrt (Const b)), Const b, Addition (Const c) (Sqrt (Sqrt (Const d))), Sqrt (Const d), Const d\}.
```

```haskell
fun radicals :: expr => expr set
where
  radicals (Const x) = {}
  radicals (Negation e) = (radicals e)
  radicals (Inverse e) = (radicals e)
  radicals (Addition e1 e2) = ((radicals e1) ∪ (radicals e2))
  radicals (Multiplication e1 e2) = ((radicals e1) ∪ (radicals e2))
  radicals (Sqrt e) = (if \|e\| < 0 then radicals e else {e} ∪ (radicals e))
```

If `r` is in `radicals` of `e` then the set `radical-sqrt` of `r` is a subset (strictly speaking) of the set `radicals` of `e`.

**lemma** radicals-expr-subset: \(r \in \text{radicals } e \implies \text{radical-sqrt } r \subseteq \text{radicals } e\)

by (induct e, auto simp add: if-split-asm)

If `x` is in `radical-sqrt` then there exists a radical expression `e` which translation is `x` (it is important to notice that this expression is not necessarily unique).

**lemma** radical-sqrt-correct-expr:

\(x \in \text{radical-sqrt } \implies (\exists e. \|e\| = x)\)

apply (rule radical-sqrt.induct)
apply auto
apply (erule Rats-induct)
apply (metis translation.simps(1))
apply (metis translation.simps(2))
apply (metis translation.simps(3))
apply (metis translation.simps(4))
apply (metis translation.simps(5))
apply (metis linorder-not-less translation.simps(6))
done

The order of an expression is the maximum number of radicals one over another occurring in a given expression. Using the example above, suppose \( expr \) is of the form: \( expr = \text{Addition}(\text{Sqrt}(\text{Addition}(\text{Const } a) \text{ Sqrt}(\text{Const } b))) (\text{Sqrt}(\text{Addition}(\text{Const } c) (\text{Sqrt}(\text{Sqrt}(\text{Const } d))))) \), where \( a, b, c \) and \( d \) are rationals and which can be translated as follows: \( \{ expr \} = \sqrt{a + \sqrt{b + \sqrt{c + \sqrt{d}}}}. \) The order of \( expr \) is \( max(2, 3) = 3. \)

fun order :: expr => nat
where
  order (Const x) = 0 |
  order (Negation e) = order e |
  order (Inverse e) = order e |
  order (Addition e1 e2) = max (order e1) (order e2) |
  order (Multiplication e1 e2) = max (order e1) (order e2) |
  order (Sqrt e) = 1 + order e

If an expression \( s \) is one of the radicals (or in radicals) of the expression \( r \), then its order is smaller (strictly speaking) then the order of \( r \).

lemma in-radicals-smaller-order:
  \( s \in \text{radicals } r \implies (order s) < (order r) \)
apply (induct r, auto)
apply (metis insert-iff insert-is-Un less-Suc-eq)
done

The following theorem is the converse of the previous lemma.

lemma in-radicals-smaller-order-contrap:
  \( (order s) \geq (order r) \implies \neg (s \in \text{radicals } r) \)
by (metis in-radicals-smaller-order order-less-irrefl)

An expression \( r \) cannot be one of its own radicals.

lemma not-in-own-radicals:
  \( \neg (r \in \text{radicals } r) \)
by (metis in-radicals-smaller-order order-less-irrefl)

If an expression \( e \) is a radical expression and it has no radicals then its translation is a rational.

lemma radicals-empty-rational: radicals \( e = \{ \} \implies \{ e \} \in \mathbb{Q} \)
by (induct e, auto)

A finite non-empty set of natural numbers has necessarily a maximum.
lemma finite-set-has-max:
    finite (s:: nat set) → s ≠ {} → ∃ k ∈ s. ∀ p ∈ s. p ≤ k
by (metis Max-ge Max-in)

There is a finite number of radicals in an expression.

lemma finite-radicals: finite (radicals e)
by (induct e, auto)

We define here a new set corresponding to the orders of each element in the set radicals of an expression expr. Using the example above, suppose expr is of the form: expr = Addition (Sqrt (Addition (Const a) Sqrt (Const b))) (Sqrt (Addition (Const c) (Sqrt (Const d))))), where a, b, c and d are rationals and which can be translated as follows: \(|expr| = \sqrt{a} + \sqrt{b} + \sqrt{c} + \sqrt{d} \). The set radicals of expr is \{Addition (Const a) Sqrt (Const b), Const b, Addition (Const c) (Sqrt (Const d)), Sqrt (Const d), Const d\}; therefore, the set order-radicals of this set is \{1, 0, 2, 1, 0\}.

fun order-radicals:: expr set => nat set
where order-radicals s = {y. ∃ x ∈ s. y = order x}

If the set of radicals of an expression e is not empty and is finite then the set order-radicals of the set of radicals of e is not empty and is also finite.

lemma finite-order-radicals:
    radicals e ≠ {} → finite (order-radicals s) →
    order-radicals (radicals e) ≠ {} ∧ finite (order-radicals (radicals e))
by simp (metis equals0I)

The following lemma states that given an expression e, if the set order-radicals of the set radicals e is not empty and is finite, then there exists a radical r of e which is of highest order among the radicals of e.

lemma finite-order-radicals-has-max:
    order-radicals (radicals e) ≠ {} →
    finite (order-radicals (radicals e)) →
    ∃ r. (r ∈ radicals e) ∧ (∀ s ∈ (radicals e). (order r ≥ order s))
using finite-set-has-max [of order-radicals (radicals e)]
by auto

This important lemma states that in an expression that has at least one radical, we can find an upmost radical r which is not radical of any other term of the expression e. It is also important to notice that this upmost radical is not necessarily unique and is not the term of highest order of the expression e. Using the example above, suppose e is of the form: e = Addition (Sqrt (Addition (Const a) Sqrt (Const b))) (Sqrt (Addition (Const c) (Sqrt (Const d))))), where a, b, c and d are rationals and which can be translated as follows: \(|e| = \sqrt{a} + \sqrt{b} + \sqrt{c} + \sqrt{d} \). The possible
The upmost radicals in this expression are Addition (Const \(a\)) (Sqrt (Const \(b\))) or Addition (Const \(c\)) (Sqrt (Sqrt (Const \(d\)))).

**Lemma upmost-radical-sqrt2:**

radicals \(e \neq \{\} \implies \exists \ r \in \text{radicals} \ e. \forall \ s \in \text{radicals} \ e. \ r \notin \text{radicals} \ s

using in-radicals-smaller-order-contrap [of \(r\) \(s\)] finite-radicals [of \(e\)] by (metis finite-order-radicals finite-order-radicals-has-max in-radicals-smaller-order-contrap)

The following 7 lemmas are used to prove the main lemma radical-sqrt-normal-form which states that if an expression \(e\) has at least one radical then it can be written in a normal form. This means that there exist three radical expressions \(a\), \(b\) and \(r\) such that \(\|e\| = \|a\| + \|b\| \times \sqrt{\|r\|}\) and the radicals of \(a\) are radicals of \(e\) but are not \(r\), and the same goes for the radicals of \(b\) and \(r\). It is important to notice that \(a\), \(b\) and \(r\) are not unique and Sqrt \(r\) is not necessarily the term of highest order.

**Lemma radical-sqrt-normal-form-sublemma:**

((\(a::real\) – \(b\)) \(*\) (\(a + b\)) = \(a \times a – b \times b\) by (simp add: field-simps)

**Lemma eq-sqrt-squared:**

(x::real) \(\geq\) \(0 \implies (\sqrt{x}) \times (\sqrt{x}) = x\) by (metis abs-of-nonneg real-sqrt-abs2 real-sqrt-mult)

**Lemma radical-sqrt-normal-form-lemma4:**

assumes \(z \geq \) \(0\) \(x \neq y \times \sqrt{z}\) shows \(1 / (x + y \times \sqrt{z}) = (x / (x + y \times \sqrt{z} ) – (y \times \sqrt{z}) / (x \times x – y \times y \times z)) / (x – y \times \sqrt{z})\)

proof –

have \(1 / (x + y \times \sqrt{z}) = ((x – y \times \sqrt{z}) / (x + y \times \sqrt{z})) / (x – y \times \sqrt{z})\)

by (auto simp add: eq-divide-imp assms)

also have \(\ldots = x / (x \times x – y \times y \times z) – (y \times \sqrt{z}) / (x \times x – y \times y \times z)\)

by (auto simp add: algebra-simps eq-sqrt-squared diff-divide-distrib assms)

finally show \(?thesis\). qed

**Lemma radical-sqrt-normal-form-lemma:**

fixes \(e::\text{expr}\)

assumes radicals \(e \neq \{\}\)

and \(\forall \ s \in \text{radicals} \ e. \ r \notin \text{radicals} \ s\)

and \(r : \text{radicals} \ e\)

shows \(\exists \ a, b. \ 0 \leq \|r\| \& \|e\| = \|a\| + \|b\| \times \sqrt{\|r\|}\) &

radicals \(a \cup \) radicals \(b \cup \) radicals \(r \subseteq\) radicals \(e\) &

\(r \notin\) radicals \(a \cup\) radicals \(b\)

(is \(\exists \ a, b. \ ?concl\ e\ a\ b\))

using assms

proof (induct \(e\)
case (Const rat) thus ?case
  by auto
next
case (Negation e)
  obtain a b
    where a2: ?concl e a b
    by (metis Negation radicals.simps(2))
hence \{Negation e\} = \{Negation a\} + \{Negation b\} * sqrt \{r\}
  by simp
thus ?case using a2
  by (metis radicals.simps(2))
next
case (Inverse e)
  obtain a b
    where ?concl e a b
    by (metis Inverse radicals.simps(3))
thus ?case
apply (case-tac \{b\} * sqrt \{r\} = \{a\})
apply simp
apply (case-tac \{a\} = 0)
apply (metis add-0-right div-by-0 mult-zero-right)
apply (rule-tac x = Multiplication (Const 1) (Inverse (Multiplication (Const 2) a)) in exI)
  apply (rule-tac x = Const 0 in exI, simp)
  apply (rule-tac x = Multiplication a (Inverse (Addition (Multiplication a a) (Negation (Multiplication (Multiplication b b) r)))) in exI)
  apply (rule-tac x = Negation (Multiplication b (Inverse (Addition (Multiplication a a) (Negation (Multiplication (Multiplication b b) r)))))) in exI)
  apply (simp add: algebra-simps not-in-own-radicals eq-diff-eq' radical-sqrt-normal-form-lemma4)
done
next
case (Addition e1 e2)
hence d1: \forall s \in radicals e1 \cup radicals e2. r \notin radicals s
  by (metis radicals.simps(4))
show ?case
proof (cases r: radicals e1 & r : radicals e2)
case True
  obtain a1 b1 a2 b2
    where ab: ?concl e1 a1 b1
    and bb: ?concl e2 a2 b2
    using Addition.hyps
    by (simp add: d1) (metis True empty-iff)
thus ?thesis
apply simp
apply (rule-tac x = Addition a1 a2 in exI)
apply (rule-tac x = Addition b1 b2 in exI)
apply (auto simp add: comm-semiring-class.distrib)
done
next
case False
      thus ?thesis
      proof (cases r: radicals e1)
      case True
      obtain a1 b1
      where 0 ≤ |r| ?concl e1 a1 b1
            using Addition.hyps
      by (auto simp: d1) (metis True empty-iff)
      thus ?thesis
      apply (rule-tac x = Addition a1 e2 in exI)
      apply (rule-tac x = a1 in exI) using False True
      apply auto
      done
    next
    case False
    obtain a2 b2
      where 0 ≤ |r| ?concl e2 a2 b2
      using Addition by (metis False Un-iff empty-iff radicals.simps(4))
    thus ?thesis
    apply (rule-tac x = Addition a2 e1 in exI)
    apply (rule-tac x = a2 in exI) using False
    apply auto
    done
  qed
qed
next
  case (Multiplication e1 e2)
  show ?case
  proof (cases r: radicals e1 & r : radicals e2)
  case True
  then obtain a1 b1 a2 b2
  where ?concl e1 a1 b1 ?concl e2 a2 b2
        using Multiplication
  by simp (metis True empty-iff)
  thus ?thesis
  apply (rule-tac x = Addition (Multiplication a1 a2) (Multiplication r (Multiplication b1 b2))) in exI
  apply (rule-tac x = Addition (Multiplication a1 b2) (Multiplication a2 b1)
        in exI)
  apply (auto simp add: not-in-own-radicals algebra-simps eq-sqrt-squared)
  done
next
  case False
  thus ?thesis
  proof (cases r: radicals e1)
  case True
  then obtain a1 b1
  where ?concl e1 a1 b1

12
using Multiplication.hyps Multiplication(4)
by auto (metis True empty-iff)
thus ?thesis
apply (rule-tac x = Multiplication a1 e2 in exI)
apply (rule-tac x = Multiplication b1 e2 in exI)
apply (simp add: algebra-simps)
by (metis False True le-supI1 radicals.simps(5))

next
case False
then obtain a2 b2 where ?concl e2 a2 b2
using Multiplication.hyps Multiplication(4) Multiplication(5)
by auto blast
thus ?thesis
apply (rule-tac x = Multiplication a2 e1 in exI)
apply (rule-tac x = Multiplication b2 e1 in exI)
apply (simp add: algebra-simps)
by (metis False le-supI2)

qed
qed

next
case (Sqrt e)
show ?case
proof (cases \{ e \} < 0)
case True thus ?thesis
using Sqrt
apply (rule-tac x = Const 0 in exI)
apply (rule-tac x = Const 0 in exI)
apply auto
done
next
case False thus ?thesis
apply (rule-tac x = Const 0 in exI)
apply (rule-tac x = Const 1 in exI) using Sqrt
apply (auto simp add: linorder-not-less)
done
qed
qed

This main lemma is essential for the remaining part of the proof.

theorem radical-sqrt-normal-form:
radicals e \neq \{\} \implies
\exists \ r \in \text{radicals}\ e. \\
\exists \ a \ b. \{e\} = \{\text{Addition} a (\text{Multiplication} b (Sqrt r))\} \wedge \{r\} \geq 0 \wedge \\
\text{radicals}\ a \cup \text{radicals}\ b \cup \text{radicals}\ r \subseteq \text{radicals}\ e \& \\
r \notin \text{radicals}\ a \cup \text{radicals}\ b \cup \text{radicals}\ r
using upmost-radical-sqrt2 [of e] radical-sqrt-normal-form-lemma
by auto (metis all-not-in-conv leD)
2.7 Important properties of the roots of a cubic equation

The following 7 lemmas are used to prove a main result about the properties of the roots of a cubic equation (cubic-root-radical-sqrt-rational) which states that assuming that \( a \) and \( b \) are rationals and that \( x \) is a radical satisfying \( x^3 + ax^2 + bx + c = 0 \) then there exists a rational root. This lemma will be used in the proof of the impossibility of trisection an angle and of duplicating a cube.

**Lemma** cubic-root-radical-sqrt-steplemma:

fixes \( P \) :: real set

assumes Nats \([\text{THEN set-mp, intro}]: \text{Nats} \subseteq P\)

and \( \text{Neg} \): \( \forall x \in P. \ -x \in P \)

and \( \text{Inv} \): \( \forall x \in P. \ x \neq 0 \rightarrow 1/x \in P \)

and \( \text{Add} \): \( \forall x \in P. \ \forall y \in P. \ x+y \in P \)

and \( \text{Mult} \): \( \forall x \in P. \ \forall y \in P. \ xy \in P \)

and \( a: (a \in P) \) and \( b: (b \in P) \) and \( c: (c \in P) \)

and \( eq0: \ z^3 + a * z^2 + b * z + c = 0 \)

and \( u: (u \in P) \)

and \( v: (v \in P) \)

and \( s: ((s * s) \in P) \)

and \( z: (z = u + v * s) \)

shows \( \exists w \in P. \ w^3 + a * w^2 + b * w + c = 0 \)

**Proof** (cases \( v * s = 0 \))

**Case** True

**Thus** \( \neg \text{thesis} \)

by (metis eq0 u z add-0-iff)

**Next**

**Case** False

hence \( sl0: v \neq 0 \)

by (metis mult-eq-0-iff)

from Add Neg have Minus \( \forall x \in P. \ \forall y \in P. \ x - y \in P \) by (simp only: diff-conv-add-uminus) blast

have \( \text{l2}: (u^3 + 3 * u * v^2 * s^2 + a * u^2 + a * v^2 * s^2 + b * u + c) + (3 * u^2 * v + v^3 * s^2 + 2 * a * u * v + b * v) * s = 0 \)

using eq0 z

by algebra

show \( \neg \text{thesis} \)

**Proof** (cases \( 3 * u^2 * v + v^3 * s^2 + 2 * a * u * v + b * v \neq 0 \))

**Case** True

hence \( s = ((3 * u^2 * v + v^3 * s^2 + 2 * a * u * v + b * v) * (1/ (3 * u^2 * v + v^3 * s^2 + 2 * a * u * v + b * v))) = - (u^3 + 3 * u * v^2 * s^2 + a * u^2 + a * v^2 * s^2 + b * u + c) * (1/ (3 * u^2 * v + v^3 * s^2 + 2 * a * u * v + b * v)) \)

using \( \text{l2} \)

by algebra

hence \( s = ((3 * u^2 * v + v^3 * s^2 + 2 * a * u * v + b * v) / (3 * u^2 * v + v^3 * s^2 + 2 * a * u * v + b * v)) = - (u^3 + 3 * u * v^2 * s^2 + a * u^2 + a * v^2 * s^2 + b * u + s) \)
c) 

\[
(1 / (3 * u^2 * v + v^3 * s^2 + 2 * a * u * v + b * v))
\]

by auto

hence \( s = - (u^3 + 3 * u * v^2 * s^2 + a * u^2 + a * v^2 * s^2 + b * u)
\)

hence \( u + c * (1 / (3 * u^2 * v + v^3 * s^2 + 2 * a * u * v + b * v)) \)

by (metis mult-1-right True divide-self-if)

hence \( u10: s = - (u * u * u + 3 * u * v * v * (s*s) + a * u * u + a * v*v * (s*s) + b * u + c) \)

by (simp add: algebra-simps power2-eq-square power3-eq-cube)

have \( (3*u*u * v + v*v*v * (s*s) + 2 * a * u * v + b * v) \in P \)

using \( a b u v s Nats Mult Add \)

by auto

hence \( u103: 1 / (3 * u * u * v + v * v*v * (s*s) + 2 * a * u * v + b * v) \in P \)

using \( Inv True \)

by auto

have \( -(u*u*u + 3 * u * v * v * (s*s) + a * u * u + a * v*v * (s*s) + b * u + c) \in P \)

using \( a b c u v s Mult Add Neg Minus Nats \)

by simp

hence \(- (u * u * u + 3 * u * v * v * (s*s) + a * u * u + a * v*v * (s*s) + b * u + c) * (1 / (3 * u * u * v + v * v*v * (s*s) + 2 * a * u * v + b * v)) \in P \)

using \( u103 Mult \)

by \(metis\)

hence \( s \in P \)

using \( u10 \)

by \(auto\)

hence \( z \in P \)

using \( z u v Mult Add \)

by \(auto\)

thus \(?thesis\)

using \( eq0 \)

by \(auto\)

next

case False

have \( (- a - 2 * u)^3 + a * (- a - 2 * u)^2 + b * (- a - 2 * u) + c = (- a - 2 * u)^3 + a * (- a - 2 * u)^2 + (- 3 * u^2 + v^2 * s^2 + 2 * a * u) \)

using \( l2 False sl0 \)

by \( algebra\)

also have \( ... = 0 \)

by \( (simp add: algebra-simps power-def)\)

finally show \(?thesis\)

by \( (metis a u Add Neg diff-conv-add-uminus mult-2)\)

qed
lemma cubic-root-radical-sqrt-steplemma-sqrt:
assumes Nats [THEN set-mp, intro]: Nats ⊆ P
and Neg: ∀x ∈ P. ¬x ∈ P
and Inv: ∀x ∈ P. x ≠ 0 → 1/x ∈ P
and Add: ∀x ∈ P. ∀y ∈ P. x + y ∈ P
and Mult: ∀x ∈ P. ∀y ∈ P. x * y ∈ P
and a: (a ∈ P) and b: (b ∈ P) and c: (c ∈ P)
and eq0: z^3 + a * z^2 + b * z + c = 0
and w: (w ∈ P)
and: (v ∈ P)
and: (s ∈ P)
and: (sPositive: s ≥ 0)
and: (z: z = u + v * sqrt s)
shows ∃ w ∈ P. w^3 + a * w^2 + b * w + c = 0
proof
have (sqrt s) * (sqrt s) ∈ P
  by (metis eq-sqrt-squared s sPositive)
thus ?thesis using cubic-root-radical-sqrt-steplemma [of P a b c z u v sqrt s]
  Neg Add Mult Inv a b c u v s eq0 z
by auto
qed

lemma cubic-root-radical-sqrt-lemma:
fixes e::expr
assumes a: a ∈ Q and b: b ∈ Q and c: c ∈ Q
and notEmpty: radicals e ≠ {}
and eq0: {e}^3 + a * {e}^2 + b * e + c = 0
shows ∃ e1. radicals e1 ⊆ radicals e & (e1)^3 + a * e1^2 + b * e1 + c = 0
proof
obtain r u v
  where hypsruv: {r} ≥ 0 r ∈ radicals e
  {e} = {Addition u (Multiplication v (Sqrt r))}
  radicals u ∪ radicals v ∪ radicals r ⊆ radicals e
  r ∉ radicals u ∪ radicals v ∉ radicals r
  using notEmpty radical-sqrt-normal-form [of e]
by blast
let ?E = {x. ∃ ex. (|ex| = x) & ((radicals ex) ⊆ (radicals e)) & (r ∉ (radicals ex))}
  have NatsE: Nats ⊆ ?E
    by (force elim: Nats-cases intro: exI[of - Const (rat-of-nat n) for n])
  have negE: ∀x ∈ ?E. ¬x ∈ ?E
    using hypsruv by (force intro: exI[of - Negation ex for ex])
  have invE: ∀x ∈ ?E. x ≠ 0 --> 1/x ∈ ?E
    using hypsruv by (force intro: exI[of - Inverse ex for ex])
  have addE: ∀x ∈ ?E. ∀y ∈ ?E. x + y ∈ ?E
using hypsrw by (force intro: exI[of - Addition ex1 ex2 for ex1 ex2])
have multE: \( \forall x \in {?E}. \forall y \in {?E}. x \ast y \in {?E} \)
  using hypsrw by (force intro: exI[of - Multiplication ex1 ex2 for ex1 ex2])
obtain ra rb rc
  where hypsra: a = of-rat ra
  and hypsrb: b = of-rat rb
  and hypsrc: c = of-rat rc
unfolding Rats-def
  by (metis Rats-cases a b c)
have a \in {?E} \& b \in {?E} \& c \in {?E} \& \{|u|\} \in {?E} \& \{|v|\} \in {?E} \& \{|r|\} \geq 0 \& \{|e|\} = \{|u|\} + \{|v|\} \ast \sqrt{|r|} 
  using a b c notEmpty hypsrw hypsra hypsrb hypsrc
with eq0 hypsrw NatsE negE invE addE multE
cubic-root-radical-sqrt-lemma [of ?E a b c e]
obtain w where w \in {?E} \& (w^3 + a \ast w^2 + b \ast w + c = 0)
  by auto
then obtain e2
  where \{|e2|\} = w radicals e2 \subseteq radicals e r \notin radicals e2
  \{|e2|\}^3 + a \ast \{|e2|\}^2 + b \ast \{|e2|\} + c = 0
  by auto
with hypsrw show ?thesis
  by (metis subset-iff-psubset-eq)

lemma cubic-root-radical-sqrt:
  assumes abc: a \in \mathbb{Q} \& b \in \mathbb{Q} \& c \in \mathbb{Q}
  shows card (radicals e) = n \Longrightarrow \{|e|\}^3 + a \ast \{|e|\}^2 + b \ast \{|e|\} + c = 0 \Longrightarrow \exists x \in \mathbb{Q}. x^3 + a \ast x^2 + b \ast x + c = 0
proof (induct n arbitrary: e rule: less-induct)
case (less n)
  thus \(?case \)
proof \(\) cases
  assume n: n = 0
  thus \(?thesis \)
  using less.prems radicals-empty-rational [of e] finite-radicals [of e]
  by (auto simp add: card-eq-0-iff n)
next
  assume n \neq 0
  hence card (radicals e) \neq 0
  using less.prems by auto
  hence radicals e \neq \{
  by (metis card.empty)
  hence \(\exists e1. \) radicals e1 \(\subseteq\) radicals e & (\{|e1|\}^3 + a \ast \{|e1|\}^2 + b \ast \{|e1|\} + c = 0)
  using abc less.prems cubic-root-radical-sqrt-lemma [of a b c e]
  by auto
  then obtain e1
  where hypsel1: radicals e1 \(\subseteq\) radicals e & (\{|e1|\}^3 + a \ast \{|e1|\}^2 + b \ast \{|e1|\} + c = 0)
+ c = 0)

by auto

hence card (radicals e1) < card (radicals e)

by (metis finite-radicals psubset-card-mono)

hence card (radicals e1) < n & a : Rats & b : Rats & c : Rats & \{e1\}^3 + a * \{e1\}^2 + b * \{e1\} + c = 0

using hypsel less.prems abc

by auto

thus \thetahyp using less.hyps [of - e1]

by auto

qed

Now we can prove the final result about the properties of the roots of a cubic equation.

theorem cubic-root-radical-sqrt-rational:

assumes a: a ∈ Q and b: b ∈ Q and c: c ∈ Q

and x: x ∈ radical-sqrt

and x-eqn: x^3 + a * x^2 + b * x + c = 0

shows c: ∃ x ∈ Q, x^3 + a * x^2 + b * x + c = 0

proof (cases a = 0)

case True

thus \thetahyp using abNotNull eq0

by auto

qed

2.8 Important properties of radicals

lemma sqrt-roots:

y^2 = x ⇒ x ≥ 0 & (sqrt (x) = y | sqrt (x) = -y)

apply (simp add: power-def)

by (metis abs-of-nonneg abs-of-nonpos real-sqrt-abs2 zero-le-mult-iff zero-le-square)

lemma radical-sqrt-linear-equation:

assumes a: a ∈ radical-sqrt

and b: b ∈ radical-sqrt

and abNotNull: ~ (a = 0 & b = 0)

and eq0: a * x + b = 0

shows x ∈ radical-sqrt

proof (cases a=0)

case True

thus \thetahyp using abNotNull eq0

by auto
next
  case False
  hence \( l0: a \neq 0 \)
    by simp
  hence \( x = -b/a \)
    using \( eq0 \) by (simp add: field-simps)
  also have \( \ldots \in \text{radical-sqrt} \)
    using \( a \) \( b \) \text{radical-sqrt} \( l0 \)
    by (metis \text{radical-sqrt} \text{intros(2)} \text{radical-sqrt-rule-division})
  finally show \(?thesis\)
qed

lemma \text{radical-sqrt-simultaneous-linear-equation:}
  assumes \( a: a \in \text{radical-sqrt} \)
  and \( b: b \in \text{radical-sqrt} \)
  and \( c: c \in \text{radical-sqrt} \)
  and \( d: d \in \text{radical-sqrt} \)
  and \( e: e \in \text{radical-sqrt} \)
  and \( f: f \in \text{radical-sqrt} \)
  and \( \text{NotNull}: \neg (a*e-b*d = 0 \& a*f - c*d = 0 \& e*c = b*f) \)
  and \( \text{eq0}: a*x + b*y = c \)
  and \( \text{eq1}: d*x + e*y = f \)
  shows \( x \in \text{radical-sqrt} \& y \in \text{radical-sqrt} \)
proof \((\text{cases} a*e - b*d = 0)\)
  case False
  hence \((a*e-b*d) * x = (e*c-b*f)\) using \( eq0 \) \( eq1 \)
    by algebra
  hence \( x: x = (c*e-b*f) / (a*e-b*d) \)
    using False by (simp add: field-simps)
  hence \((a*e-b*d) * y = (a*f - d*c)\) using \( eq0 \) \( eq1 \)
    by algebra
  hence \( y: y = (a*f - d*c)/(a*e-b*d) \)
    using False by (simp add: field-simps)
  have \( \text{ae-rad}: (a*e-b*d) \in \text{radical-sqrt} \)
    using \( a \) \( e \) \( b \) \( d \) \text{radical-sqrt.simps}
    by (metis \text{radical-sqrt} \text{intros(5)} \text{radical-sqrt-rule-subtraction})
  hence \(((e*c-b*f)/ (a*e-b*d)) \in \text{radical-sqrt} \ ((a*f - d*c) / (a*e-b*d)) \in \text{radical-sqrt} \)
    using False \( a \) \( b \) \( c \) \( d \) \( e \) \( f \) by (auto intro!: \text{radical-sqrt} \text{intros(5)} \text{radical-sqrt-rule-division} \text{radical-sqrt-rule-subtraction})
    thus \(?thesis\)
      by (simp add: \( x \) \( y \))
next
  case True
  hence \((a*e-b*d) * x = (e*c-b*f) \) \( (a*e-b*d) * y = (a*f - d*c)\) using \( eq0 \) \( eq1 \)
    by algebra
  thus \(?thesis\) using \( \text{NotNull} \) \( True \)
lemma radical-sqrt-quadratic-equation:
assumes a: a ∈ radical-sqrt
  and b: b ∈ radical-sqrt
  and c: c ∈ radical-sqrt
  and eq0: a*x^2+b*x+c =0
  and NotNull: ¬ (a = 0 & b = 0 & c = 0)
shows x ∈ radical-sqrt

proof (cases a=0)
case True
  have ¬ (b = 0 & c = 0)
    by (metis True NotNull)
  thus ?thesis
    using b c radical-sqrt-linear-equation [of b c x]
    by (metis True add-0 eq0 mult-zero-left)
next
case False
  hence (2*a*x+b)^2 = 4*a*(- c)+b^2 using eq0
    by algebra
  hence (b^2 - 4*a*c)≥ 0 & (sqrt ((b^2 - 4*a*c)) = (2*a*x+b) | sqrt ((b^2 -
    4*a*c)) = -(2*a*x+b))
    using sqrt-roots [of 2*a*x+b b^2 - 4*a*c]
    by auto
  hence l12: b^2 - 4*a*c ≥ 0 & ((-b + sqrt (b^2 - 4*a*c)) / (2*a) = x |
    (b - sqrt (b^2 - 4*a*c)) / (2*a) = x)
    using False
    by auto
  have 4*a*c ∈ radical-sqrt
    using a c radical-sqrt.simps
    by (metis Rats-number-of radical-sqrt.intros(1) radical-sqrt.intros(5))
  hence b^2 - 4*a*c ∈ radical-sqrt using a b c
    by (metis power2-eq-square radical-sqrt.intros(5) radical-sqrt-rule-subtraction)
  hence l22: sqrt (b^2 - 4*a*c) ∈ radical-sqrt
    using l12
    by (metis radical-sqrt.intros(6))
  hence l23: (-b + sqrt (b^2 - 4*a*c)) / (2*a) ∈ radical-sqrt
    using b a False
    apply (simp add: algebra-simps)
    apply (metis radical-sqrt-rule-division radical-sqrt-rule-subtraction double-zero-sym
    mult-2-right mult-2-right radical-sqrt.intros(4))
    done
  have (-b - sqrt (b^2 - 4*a*c)) / (2*a) ∈ radical-sqrt
    using a b False l22
    by (metis div-by-0 mult-2 radical-sqrt.intros(2) radical-sqrt.intros(4) radical-sqrt-rule-division
    radical-sqrt-rule-subtraction)
  thus ?thesis

by simp
qed
by (metis l12 l23)

qed

lemma radical-sqrt-simultaneous-linear-quadratic:
  assumes a: a ∈ radical-sqrt
  and b: b ∈ radical-sqrt
  and c: c ∈ radical-sqrt
  and d: d ∈ radical-sqrt
  and e: e ∈ radical-sqrt
  and f: f ∈ radical-sqrt
  and NotNull: ¬(d=0 & e=0 & f=0)
  and eq1: ((x-a)^2 + (y-b)^2 = c)
  and eq1: d*x + e*y = f

  shows x ∈ radical-sqrt & y ∈ radical-sqrt

  proof (cases d=0 & e=0)
  
  case False
  hence l10: (a^2 + d^2) * x^2 + (2*a*b*d - 2*a*e^2 - 2*d*f)*x + (a^2 + e^2 + f^2 - 2*e*b*f + b^2*e^2 - e^2*c) = 0
  using eq0 eq1
  by algebra
  have l12: ¬(a^2 + d^2 = 0 & 2*a*b*d - 2*a*e^2 - 2*d*f = 0 & a^2 * e^2 + f^2 - 2*e*b*f + b^2*e^2 - e^2*c = 0)
  using False power-def
  by auto
  have l13: (a^2 + d^2) ∈ radical-sqrt
  using e d
  by (metis power2-eq-square radical-sqrt.intros(4) radical-sqrt.intros(5))
  have 2: 2 ∈ radical-sqrt
  by (auto intro: radical-sqrt.intros)
  have sl1: (2*e*b*d) ∈ radical-sqrt
  using e b d
  by (metis (lifting) mult-2 radical-sqrt.intros(4) radical-sqrt.intros(5))
  hence sl2: (¬ 2*a*e^2) ∈ radical-sqrt using radical-sqrt.intros
  by (simp add: field-simps (metis mult-2 power2-eq-square a c)
  have (¬ 2*d*f) ∈ radical-sqrt using radical-sqrt.intros
  using d f by (auto intro: radical-sqrt.intros simp add: power2-eq-square)
  hence sl4: ((2*e*b*d) + (- 2*a*e^2) + (- 2*d*f)) ∈ radical-sqrt
  using sl1 sl2
  by (metis radical-sqrt.intros(4))
  have sl5: 2*e*b*d - 2*a*e^2 - 2*d*f = (2*e*b*d) + (- 2*a*e^2) + (- 2*d*f)
  by auto
  hence l14: (2*e*b*d - 2*a*e^2 - 2*d*f) ∈ radical-sqrt
  using sl4
by \texttt{metis}

have sl6: \((a^2 \ast c^2) \in \text{radical-sqrt}\) using \(a \ast c\) by \((\text{auto intro: radical-sqrt.intros simp add: power2-eq-square})\)

have sl7: \((f^2) \in \text{radical-sqrt}\) using \(f\) by \((\text{auto intro: radical-sqrt.intros simp add: power2-eq-square})\)

have sl8: \((-2 \ast c \ast b \ast f) \in \text{radical-sqrt}\) using \(e \ast b \ast f\) by \((\text{auto intro: radical-sqrt.intros simp add: power2-eq-square})\)

have sl9: \((b^2 \ast c^2) \in \text{radical-sqrt}\) using \(b \ast c\) by \((\text{auto intro: radical-sqrt.intros simp add: power2-eq-square})\)

have sl10: \((-c \ast c^2) \in \text{radical-sqrt}\) using \(c \ast c\) by \((\text{auto intro: radical-sqrt.intros simp add: power2-eq-square})\)

have sl6: \((a^2 \ast c^2 + f^2 + (-2 \ast e \ast b \ast f) + b^2 \ast e^2 + (-c \ast c^2)) \in \text{radical-sqrt}\)

using \(a \ast b \ast c \ast sl6\) sl7 sl8 sl9 sl10

by \((\text{metis (hide-lams, no-types) power2-eq-square radical-sqrt.intros(\#4)})\)

have \((a^2 \ast c^2 + f^2 - 2 \ast e \ast b \ast f + b^2 \ast e^2 - c^2 \ast c = a^2 \ast c^2 + f^2 + (-2 \ast e \ast b \ast f) + b^2 \ast e^2 + (-c \ast e^2))\) by auto

hence \((a^2 \ast c^2 + f^2 - 2 \ast e \ast b \ast f + b^2 \ast e^2 - c^2 \ast c) \in \text{radical-sqrt}\) using sl6

by \texttt{metis}

hence \(x: x \in \text{radical-sqrt}\)

using \(\text{radical-sqrt-quadratic-equation} \ [of \ e^2 + d^2 \ast 2 \ast e \ast b \ast d - 2 \ast a \ast e^2 - 2 \ast d \ast f \ast a^2 + e^2 + f^2 - 2 \ast e \ast b \ast f + b^2 \ast e^2 - e^2 \ast c \ast x] \ l13 \ l14 \ l12 \ l10\)

by auto

have \(l18\): \(e \ast y + (d \ast x - f) = 0\)

using eq1 by auto

hence \(y: y \in \text{radical-sqrt}\)

using \(e \ast d \ast f \ast x\ False\)

\textbf{proof} \((\text{cases } e = 0)\)

case True

hence \(l22\): \(1 \ast y^2 + (-2 \ast b) \ast y + (b^2 + (x - a)^2 - c) = 0\)

using eq0 by \texttt{algebra}

have \(l24\): \(1 \in \text{radical-sqrt}\)

by \((\text{metis Rats-1 radical-sqrt.intros(\#1)})\)

have \(l25\): \((-2 \ast b) \in \text{radical-sqrt}\)

using \(b\)

by \((\text{metis minus-mult-commute mult-2 radical-sqrt.intros(\#2) radical-sqrt.intros(\#4)})\)

have \(l26\): \((b^2 + (x - a)^2 - c) \in \text{radical-sqrt}\)

using \(a \ast b \ast c \ast x\)

by \((\text{auto intro: radical-sqrt.intros radical-sqrt-rule-subtraction simp add: power2-eq-square})\)

thus \texttt{thesis}

using \(\text{radical-sqrt-quadratic-equation} \ [of \ 1::real - 2 \ast b \ast b^2 + (x - a)^2 - c \ast y] \ l22 \ l24 \ l25 \ l26\)

by auto

next
\textbf{case} False
\begin{itemize}
\item \textbf{hence} l29: \(\neg (e = 0 \& d \times x - f = 0)\)
\item by simp
\item \textbf{have} \((d \times x - f) \in \text{radical-sqrt}\)
\item by (metis radical-sqrt.intros(5) radical-sqrt-rule-subtraction)
\item \textbf{thus} \(\neg \)
\item \textbf{using} radical-sqrt-linear-equation [of \(d \times x - f\) \(d \times f\) l18 l29]
\item by auto
\end{itemize}
\textbf{qed}
\textbf{show} \(\neg\)
\textbf{by} (metis \(x\) \(y\))
\textbf{qed}

\textbf{lemma} \texttt{radical-sqrt-simultaneous-quadratic-quadratic}:
\begin{itemize}
\item \textbf{assumes} \(a\) \(\in\) \text{radical-sqrt}\n\item \(b\) \(\in\) \text{radical-sqrt}\n\item \(c\) \(\in\) \text{radical-sqrt}\n\item \(d\) \(\in\) \text{radical-sqrt}\n\item \(e\) \(\in\) \text{radical-sqrt}\n\item \(f\) \(\in\) \text{radical-sqrt}\n\item \(\text{NotEqual}\) \(\neg (a = d \& b = e \& c = f)\)
\item \(\text{eq0}\) \((x - a)^2 + (y - b)^2 = c\)
\item \(\text{eq1}\) \((x - d)^2 + (y - e)^2 = f\)
\end{itemize}
\textbf{shows} \(x \in \text{radical-sqrt} \& y \in \text{radical-sqrt}\)
\textbf{proof} –
\begin{itemize}
\item \textbf{have} \((x^2 - 2 \times a \times x + a^2 + y^2 - 2 \times y \times b + b^2) - (x^2 - 2 \times d \times x + d^2 + y^2 - 2 \times y \times e + e^2) = (c - f)\)
\item \textbf{using} eq0 eq1
\item by (simp add: algebra-simps power-def)
\item \textbf{hence} l4: \((2 \times d - 2 \times a) \times x + (2 \times e - 2 \times b) \times y + (b^2 - e^2 + a^2 - d^2 + f - c) = 0\)
\item \textbf{by} algebra
\item \textbf{hence} l6: \((a - d)^2 - 2 \times a \times x + (e - b)^2 - 2 \times y \times b + b^2\) - \((c - f) = 0\)
\item \textbf{using} \(\text{NotEqual}\)
\item \textbf{by} algebra
\item \textbf{have} l7: \((2 \times d - 2 \times a) \in \text{radical-sqrt}\)
\item \textbf{by} (metis a d mult-2 radical-sqrt.intros(4) radical-sqrt-rule-subtraction)
\item \textbf{have} l8: \((2 \times e - 2 \times b) \in \text{radical-sqrt}\)
\item \textbf{by} (metis b e mult-2 radical-sqrt.intros(4) radical-sqrt-rule-subtraction)
\item \textbf{have} be-rad: \((b^2 - e^2) \in \text{radical-sqrt}\)
\item \textbf{by} (metis b e power2-eq-square radical-sqrt.intros(5) radical-sqrt-rule-subtraction)
\item \textbf{have} ad-rad: \((a^2 - d^2) \in \text{radical-sqrt}\)
\item \textbf{by} (metis a d power2-eq-square radical-sqrt.intros(5) radical-sqrt-rule-subtraction)
\item \textbf{have} \((f - c) \in \text{radical-sqrt}\)
\item \textbf{by} f c
\item \textbf{using} \(\text{radical-sqrt-rule-subtraction}\)
\item \textbf{hence} \((b^2 - e^2) + (a^2 - d^2) + (f - c)) \in \text{radical-sqrt}\)
using radical-sqrt.intros
by (metis be-rad ad-rad)
thus ?thesis
using radical-sqrt-simultaneous-linear-quadratic [of a b c (2*d - 2*a) (2*e - 2*b) - ((b^2 - e^2) + (a^2 - d^2) + (f - c)) x y] l7 l8 l6 l4 a b c d e f NotEqual eq0 eq1
by simp
qed

2.9 Important properties of geometrical points which coordinates are radicals

lemma radical-sqrt-line-line-intersection:
assumes absA: (abscissa (A)) ∈ radical-sqrt
and ordA: (ordinate A) ∈ radical-sqrt
and absB: (abscissa B) ∈ radical-sqrt
and ordB: (ordinate B) ∈ radical-sqrt
and absC: (abscissa C) ∈ radical-sqrt
and ordC: (ordinate C) ∈ radical-sqrt
and absD: (abscissa D) ∈ radical-sqrt
and ordD: (ordinate D) ∈ radical-sqrt
and notParallel: ¬ (parallel A B C D)
and isIntersec: is-intersection X A B C D
shows (abscissa X) ∈ radical-sqrt & (ordinate X) ∈ radical-sqrt

proof-
have l2: (abscissa A - abscissa X) * (ordinate A - ordinate B) = (ordinate A - ordinate X) * (abscissa A - abscissa B) & (abscissa C - abscissa X) * (ordinate C - ordinate D) = (ordinate C - ordinate X) * (abscissa C - abscissa D)
using isIntersec is-intersection-def collinear-def parallel-def
by auto
by (simp add: algebra-simps)
have l6: (¬ (ordinate C - ordinate D)) * abscissa X + (abscissa C - abscissa D) * ordinate X = (¬ abscissa C * (ordinate C - ordinate D) + ordinate C * (abscissa C - abscissa D))
using l2
by (simp add: algebra-simps)
have sl1: (¬ (ordinate A - ordinate B)) ∈ radical-sqrt
by (metis ordA ordB minus-diff-eq radical-sqrt-rule-subtraction)
have sl2: (abscissa A - abscissa B) ∈ radical-sqrt
by (metis absA absB radical-sqrt-rule-subtraction)
have sl3: (¬ abscissa A * (ordinate A - ordinate B) + ordinate A * (abscissa A - abscissa B)) ∈ radical-sqrt
using absA ordA ordB absB
by (metis diff-cone-add-uminus radical-sqrt.intros(2) radical-sqrt.intros(4) radical-sqrt.intros(5))
have sl4: (¬ (ordinate C - ordinate D)) ∈ radical-sqrt
by (metis ordC ordD minus-diff-eq radical-sqrt-rule-subtraction)
have sl5: \((\text{abscissa } C - \text{abscissa } D) \in \text{radical-sqrt}\)

by (metis absC absD radical-sqrt-rule-subtraction)

have sl6: \((- \text{abscissa } C * (\text{ordinate } C - \text{ordinate } D) + \text{ordinate } C * (\text{abscissa } C - \text{abscissa } D)) \in \text{radical-sqrt}\)

using absC ordC absD ordD

by (metis diff-cone-add-uminus radical-sqrt.intros(2) radical-sqrt.intros(4) radical-sqrt.intros(5))

have \((- (\text{ordinate } A - \text{ordinate } B)) * (\text{abscissa } C - \text{abscissa } D) \neq (\text{abscissa } A - \text{abscissa } B) * (- (\text{ordinate } C - \text{ordinate } D))\)

using notParallel parallel-def

by (simp add: algebra-simps)

thus \(?thesis\)

using radical-sqrt-simultaneous-linear-equation \[acf - (\text{ordinate } A - \text{ordinate } B)\]

\((\text{abscissa } A - \text{abscissa } B) - \text{abscissa } A * (\text{ordinate } A - \text{ordinate } B) + \text{ordinate } A\)

\((- (\text{ordinate } A - \text{ordinate } B)) * (\text{abscissa } C - \text{abscissa } D)\)

\(\text{abscissa } X \text{ ordinate } X\)

hence \(\text{absA ordA absB ordB absC ordC absD ordD l4 sl1 sl2 sl3 sl4 sl5 sl6 l6}\)

by simp

qed

lemma radical-sqrt-line-circle-intersection:

assumes absA: \((\text{abscissa } A) \in \text{radical-sqrt}\) and ordA: \((\text{ordinate } A) \in \text{radical-sqrt}\)

and absB: \((\text{abscissa } B) \in \text{radical-sqrt}\) and ordB: \((\text{ordinate } B) \in \text{radical-sqrt}\)

and absC: \((\text{abscissa } C) \in \text{radical-sqrt}\) and ordC: \((\text{ordinate } C) \in \text{radical-sqrt}\)

and absD: \((\text{abscissa } D) \in \text{radical-sqrt}\) and ordD: \((\text{ordinate } D) \in \text{radical-sqrt}\)

and absE: \((\text{abscissa } E) \in \text{radical-sqrt}\) and ordE: \((\text{ordinate } E) \in \text{radical-sqrt}\)

and notEqual: \(A \neq B\)

and colin: \(\text{collinear } A X B\)

and eqDist: \((\text{distance } C X = \text{distance } D E)\)

shows \((\text{abscissa } X) \in \text{radical-sqrt} \& (\text{ordinate } X) \in \text{radical-sqrt}\)

proof –

have l3: \((- (\text{ordinate } A - \text{ordinate } B)) * \text{abscissa } X + (\text{abscissa } A - \text{abscissa } B) * \text{ordinate } X = (- \text{abscissa } A * (\text{ordinate } A - \text{ordinate } B) + \text{ordinate } A * (\text{abscissa } A - \text{abscissa } B))\)

using colin unfolding collinear-def parallel-def

by algebra

have sqrt \((\text{abscissa } X - \text{abscissa } C)^2 + (\text{ordinate } X - \text{ordinate } C)^2) =\)

sqrt \((\text{abscissa } D - \text{abscissa } E)^2 + (\text{ordinate } D - \text{ordinate } E)^2)\)

using eqDist distance-def

by (metis (no-types) minus-diff-eq point-abscissa-diff point-dist-def point-ordinate-diff point-diff power2-minus)

hence l6: \((\text{abscissa } X - \text{abscissa } C)^2 + (\text{ordinate } X - \text{ordinate } C)^2 =\)

\((\text{abscissa } D - \text{abscissa } E)^2 + (\text{ordinate } D - \text{ordinate } E)^2)\)

by auto

have l8: \((- (\text{ordinate } A - \text{ordinate } B) = 0 \& (\text{abscissa } A - \text{abscissa } B) =\)

0 \& (- \text{abscissa } A * (\text{ordinate } A - \text{ordinate } B) + \text{ordinate } A * (\text{abscissa } A - \text{abscissa } B)) = 0)\)

using notEqual unfolding point-eq-iff
\begin{verbatim}
by auto
have sl1: (\- \ordinates A \- \ordinates B)) \in \sqrt
  by (metis \ordinates A \ordinates B \ordinates eq \sqrt-rule-subtraction)
have sl2: (\ordinates A \- \ordinates B) \in \sqrt
  by (metis \ordinates A \ordinates B \sqrt-rule-subtraction)
have sl3: (\- \ordinates A \times \ordinates B) \+ \ordinates A \times (\ordinates A \- \ordinates B)) \in \sqrt
  by (metis \ordinates A \ordinates B \ordinates \+ \sqrt-rule-subtraction)

\textbf{thus} \?thesis

\textbf{using} \sqrt-simultaneous-linear-quadratic
\begin{aligned}
  & (\ordinates D \- \ordinates E) \times 2 \+ (\ordinates D \- \ordinates E) \times 2 \\
  & \- (\ordinates A \- \ordinates B) \ordinates A \- \ordinates B \\
  & \- \ordinates A \times (\ordinates A \- \ordinates B) \+ \ordinates A \times (\ordinates A \- \ordinates B)
\end{aligned}

\textbf{thus} \?thesis

\textbf{using} \sqrt-circle-circle-intersection:
\begin{aligned}
  & (\ordinates D \- \ordinates E) \times 2 \+ (\ordinates D \- \ordinates E) \times 2 \\
  & \- (\ordinates A \- \ordinates B) \ordinates A \- \ordinates B \\
  & \- \ordinates A \times (\ordinates A \- \ordinates B) \+ \ordinates A \times (\ordinates A \- \ordinates B)
\end{aligned}

\textbf{shows} (\ordinates A \times \sqrt \& (\ordinates X) \times \sqrt)

\textbf{proof} -
\begin{aligned}
  \textbf{have} \sqrt ((\ordinates A \- \ordinates A) \times 2 \+ (\ordinates X \- \ordinates A) \times 2) = \sqrt
  ((\ordinates B \- \ordinates C) \times 2 \+ (\ordinates B \- \ordinates C) \times 2) \\
  \textbf{by} (metis \no-types \ordinates eq \sqrt-rule-subtraction)
\end{aligned}
\end{verbatim}
sqrt ((absissa E - absissa F)^2 + (ordinate E - ordinate F)^2)
by (metis (no-types) eqDist1 distance-def minus-diff-eq point-abscissa-diff point-dist-def point-ordinate-diff power2-minus)
hence l3bis: (absissa X - absissa D)^2 + (ordinate X - ordinate D)^2 =
(absissa E - absissa F)^2 + (ordinate E - ordinate F)^2
by auto
have l4: ¬ (absissa A = absissa D & ordinate A = ordinate D)
by (metis point-eq-iff notEqual eqDist0 eqDist1)
have (ordinate B - ordinate C) ∈ radical-sqrt
by (metis ordB ordC radical-sqrt-rule-subtraction)
hence sl3: ((absissa B - absissa C)^2 + (ordinate B - ordinate C)^2) ∈ radical-sqrt
by (auto intro: radical-sqrt.intros simp add: power2-eq-square)

2.10 Definition of the set of constructible points

inductive-set constructible :: point set
where
(M ∈ points ∧ (absissa M) ∈ Q ∧ (ordinate M) ∈ Q) → M ∈ constructible|
(A ∈ constructible ∧ B ∈ constructible ∧ C ∈ constructible ∧ D ∈ constructible
∧ ¬ parallel A B C D ∧ is-intersection M A B C D) → M ∈ constructible|
(A ∈ constructible ∧ B ∈ constructible ∧ C ∈ constructible ∧ D ∈ constructible
∧ E ∈ constructible ∧ ¬ A = B ∧ collinear A M B ∧ distance C M = distance D
\( E \) \( \implies \) \( M \in \text{constructible} \)

\( (A \in \text{constructible} \land B \in \text{constructible} \land C \in \text{constructible} \land D \in \text{constructible} \land E \in \text{constructible} \land F \in \text{constructible} \land \neg (A = D \land \text{distance } B C = \text{distance } E F) \land \text{distance } A M = \text{distance } B C \land \text{distance } D M = \text{distance } E F) \implies M \in \text{constructible} \)

### 2.11  An important property about constructible points: their coordinates are radicals

**lemma** `constructible-radical-sqrt`:

- **assumes** \( h : M \in \text{constructible} \)
- **shows** \((\text{abscissa } M) \in \text{radical-sqrt} \& (\text{ordinate } M) \in \text{radical-sqrt}\)
- **apply** (rule `constructible.induct`)
- **apply** (metis `assms`)
- **apply** (metis `radical-sqrt.intros(1)`)
- **apply** (metis `radical-sqrt-line-line-intersection`)
- **apply** (metis `radical-sqrt-line-circle-intersection`)
- **apply** (metis `radical-sqrt-circle-circle-intersection`)
- **done**

### 2.12  Proving the impossibility of duplicating the cube

**lemma** `impossibility-of-doubling-the-cube-lemma`:

- **assumes** \( x : x \in \text{radical-sqrt} \)
- **and** \( x\text{-eqn}: x^3 = 2 \)
- **shows** \( \text{False} \)

**proof**

- **have** \( \exists x \in \text{Rats}. x^3 + 0 \ast x^2 + 0 \ast x + (-2) = (0::\text{real}) \)
  - **using** \( x \text{-eqn cubic-root-radical-sqrt-rational} [\text{of } 0 0 -2] \)
  - **by** auto
- **then obtain** \( y::\text{real} \) **where** \( \text{hypsy}: y \in \text{Rats} \& y^3 = 2 \)
  - **by** (simp only: `left-minus mult-zero-left add-0-right real-add-minus-iff`) auto
- **then obtain** \( r \) **where** \( \text{hypsr}: y = \text{of-rat } r \)
  - **unfolding** `Rats-def`
  - **by** (metis `Rats-cases hypsy`
- **hence** \( \exists! p. r = \text{Fract } (\text{fst } p) (\text{snd } p) \& \text{snd } p > 0 \& \text{coprime } (\text{fst } p) (\text{snd } p) \)
  - **by** (metis `quotient-of-unique`
- **then obtain** \( p q \) **where** \( \text{hyps}: r = \text{Fract } p q \& p q > 0 \& \text{coprime } p q \)
  - **by** auto
- **have** \( l6: r^3 = 2 \)
  - **by** (metis `lifting` `hypsy` `of-rat-eq-iff` `of-rat-numeral-eq` `of-rat-power`)
- **have** \( l7: r^3 = \text{Fract } (p^3) (q^3) \)
  - **using** `hyps` **by** (simp add: `power3-eq-cube`
- **have** \( l8: q^3 > 0 \& \text{coprime } (p^3) (q^3) \)
  - **using** `hyps` **by** simp
- **have** \( \text{Fract } (p^3) (q^3) = 2 \)
  - **using** `l6 l7`
  - **by** auto
- **hence** \( \text{Fract } (p^3) (q^3) = \text{Fract } 2 1 \)
by (metis rat-number-expand(3))
hence l12: \( p \cdot 3 = q \cdot 3 \cdot 2 \) using hypsp
by (simp add: eq-rat)
hence even \( (p \cdot 3) \)
by (auto intro: dvdI)
then have even \( p \)
by auto
then have \( 8 \text{ dvd } p \cdot 3 \)
by (auto simp add: dvd-def power-def)
then have even \( (q \cdot 3) \)
by (auto simp add: dvd-def)
then have even \( q \)
by auto
with \( \langle \text{even } p \rangle \) have \( 2 \text{ dvd gcd } p \ q \)
by (rule gcd-greatest)
with \( \langle \text{coprime } p \ q \rangle \) show False by simp
qed

**Theorem impossibility-of-doubling-the-cube:**
\( x^3 = 2 \implies (\text{Point } x 0) \notin \text{constructible} \)
by (metis abscissa.simps constructible-radical-sqrt impossibility-of-doubling-the-cube-lemma)

### 2.13 Proving the impossibility of trisecting an angle

**Lemma impossibility-of-trisecting-pi-over-3-lemma:**
assumes \( x: x \in \text{radical-sqrt} \)
and \( x-eqn: x^3 - 3 \cdot x - 1 = 0 \)
shows False
proof
have \( \exists x \in \text{Rats}. \ x^3 + (-3) \cdot x = (1::real) \)
using x-eqn cubic-root-radical-sqrt-rational [of 0 - 3 - 1] x
by force
then obtain \( y :: \text{real where hypsy: } y \in \text{Rats} \land y^3 - 3 \cdot y - 1 = 1 \) by auto
then obtain \( r \) where \( \text{hypsrt: } y = \text{of-rat } r \)
by (metis Rats-cases)
then obtain \( p \) where \( \text{hypsps: } r = \text{Fract } (\text{fst } p) (\text{snd } p) \land \text{snd } p > 0 \& \text{coprime } (\text{fst } p) (\text{snd } p) \)
using quotient-of-unique hypsy
by blast
have r3eq: \( r^3 - 3 \cdot r = 1 \)
using hypsy hypsrt [[hypsubst-thin = true]]
by auto (metis (hide-lams, no-types) of-rat-1 of-rat-diff of-rat-eq iff of-rat-mult of-rat-numeral-eq of-rat-power)
have l7: \( (\text{snd } p)^{3} > 0 \& \text{coprime } ((\text{fst } p)^{3}) ((\text{snd } p)^{3}) \)
using hypsp by simp
have \( r^3 = \text{Fract } ((\text{fst } p)^{3}) ((\text{snd } p)^{3}) \)
proof
then have Fract $((f s t\ p)^3)((s n d\ p)^3) - (Fract (3\ (f s t\ p)) (s n d\ p)) = 1$
using $r\ 3eq\ h y p s p$
by (simp add: $F r a c t$-of-int-quotient)
then have $10:\ Fract ((f s t\ p)^3)((s n d\ p)^3) - Fract (3\ (f s t\ p)\ (s n d\ p)^2)$
$((s n d\ p)^3) = 1$
using $h y p s p$
by (simp add: power-def algebra-simps $F r a c t$-of-int-quotient)

have $F r a c t (((f s t\ p)^3 - (3\ (f s t\ p)\ (s n d\ p)^2))((s n d\ p)^3) =$
$Fract (((f s t\ p)^3 - (3\ (f s t\ p)\ (s n d\ p)^2))\ (s n d\ p)^3)\ (((s n d\ p)^3)\ (s n d\ p)^3) =$
using $17$

mult-rat-cancel [of $(s n d\ p)^3 ((f s t\ p)^3 - (3\ (f s t\ p)\ (s n d\ p)^2))\ (s n d\ p)^3]$
by (auto simp add: algebra-simps)
also have $... = Fract 1\ 1$
by (metis $17\\\ 10$ one-rat diff-rat mult-neg-pos not-square-less-zero int-distrib(3))
finally have $(f s t\ p)^3 - 3\ (f s t\ p)\ (s n d\ p)^2 = (s n d\ p)^3$
using $h y p s p$
by (simp add: eq-rat)
hence $(f s t\ p)\ (f s t\ p)^2 - 3\ (s n d\ p)^2) = (s n d\ p)^3$
$(s n d\ p)\ (s n d\ p)^2 + 3\ (f s t\ p)\ (s n d\ p)) = (f s t\ p)^3$
by (auto simp add: power-def algebra-simps)
hence $(f s t\ p)\ \ d e d ((s n d\ p)^3)\ (s n d\ p)\ \ d e d ((f s t\ p)^3)$
apply (auto simp add: ded-def)
apply (rule-tac $x = (f s t\ p)^2 - 3\ (s n d\ p)^2)\ in\ e x l$
apply (rule-tac $[2] x = (s n d\ p)^2 + 3\ (f s t\ p)\ (s n d\ p)\ in\ e x l$
apply auto
done
moreover have $c o p r i m e (f s t\ p)\ (s n d\ p)^3\ c o p r i m e (f s t\ p)^3\ (s n d\ p)$
using $h y p s p$ by auto
ultimately have $i s - u n i t (f s t\ p)\ i s - u n i t (s n d\ p)$
using $c o p r i m e - c o m m o n - d i v i s o r$ [of $f s t\ p\ s n d\ p^3\ f s t\ p$]
$c o p r i m e - c o m m o n - d i v i s o r$ [of $f s t\ p^3\ s n d\ p\ s n d\ p$]
by auto
with $h y p s p$ have $f s t\ p = 1\ \ \ \ o r\ f s t\ p = -1\ s n d\ p = 1$
by auto
with $h y p s p$ have $r = 1\ \ \ |\ r = -1$
by (auto simp add: rat-number-collapse)
with $r\ 3eq$ show $F a l s e$
by (auto simp add: power-def algebra-simps)

qed

theorem impossibility-of-trisecting-angle-$pi$-over-$3$:
Point $(c o s (p i/9))$ $\notin$ constructible

proof
have $c o s (3\ (p i/9)) = 4\ (c o s (p i/9))^3 - 3\ \ c o s (p i/9)$
using $c o s - t r e b l e - c o s$ [of $p i/9$
by auto
hence \( \frac{1}{2} = 4 \times (\cos (\pi/9))^3 - 3 \times \cos (\pi/9) \)
by \(\text{simp add: cos-60}\)
hence \(8 \times (\cos (\pi/9))^3 - 6 \times \cos (\pi/9) - 1 = 0\)
by \(\text{simp add: algebra-simps}\)
hence \((2 \times \cos (\pi/9))^3 - 3 \times (2 \times \cos (\pi/9)) - 1 = 0\)
by \(\text{simp add: algebra-simps power-def}\)
hence \(\sim (2 \times \cos (\pi/9)) \in \text{radical-sqrt}\)
by \(\text{metis impossibility-of-trisecting-pi-over-3-lemma}\)
hence \(\sim (\cos (\pi/9)) \in \text{radical-sqrt}\)
by \(\text{metis divide-self-if mult-zero-right one-add-one radical-sqrt.intros(4) radical-sqrt.intros(5) radical-sqrt-rule-division}\)
thus \(?\text{thesis}\)
by \(\text{metis absissa.simps constructible-radical-sqrt}\)
qed
end

References