The Transcendence of π

Manuel Eberl

June 16, 2019

Abstract

This entry shows the transcendence of π based on the classic proof using the fundamental theorem of symmetric polynomials first given by von Lindemann in 1882, but the mostly formalisation follows the version by Niven [3]. The proof reuses much of the machinery developed in the AFP entry on the transcendence of e.

Contents

1 Preliminary facts

theory *Pi-Transcendental-Polynomial-Library*
imports *HOL−Computational-Algebra.Computational-Algebra*
begin

lemma *Ints-sum*: $(\bigwedge x.\ x \in A \Longrightarrow f\ x \in \mathbb{Z}) \Longrightarrow sum\ f\ A \in \mathbb{Z}$
⟨*proof*⟩

lemma *Ints-prod*: $(\bigwedge x.\ x \in A \Longrightarrow f\ x \in \mathbb{Z}) \Longrightarrow prod\ f\ A \in \mathbb{Z}$
⟨*proof*⟩

lemma *sum-in-Rats* [*intro*]: $(\bigwedge x.\ x \in A \Longrightarrow f\ x \in \mathbb{Q}) \Longrightarrow sum\ f\ A \in \mathbb{Q}$
⟨*proof*⟩

lemma *prod-in-Rats* [*intro*]: $(\bigwedge x.\ x \in A \Longrightarrow f\ x \in \mathbb{Q}) \Longrightarrow prod\ f\ A \in \mathbb{Q}$
⟨*proof*⟩

lemma *poly-cnj*: $cnj\ (poly\ p\ z) = poly\ (map\text{-}poly\ cnj\ p)\ (cnj\ z)$
⟨*proof*⟩

lemma *poly-cnj-real*:
assumes $\bigwedge n.\ poly.coeff\ p\ n \in \mathbb{R}$
shows $cnj\ (poly\ p\ z) = poly\ p\ (cnj\ z)$
⟨*proof*⟩

lemma *real-poly-cnj-root-iff*:
assumes $\bigwedge n.\ poly.coeff\ p\ n \in \mathbb{R}$
shows $poly\ p\ (cnj\ z) = 0 \longleftrightarrow poly\ p\ z = 0$
⟨*proof*⟩

lemma *coeff-pcompose-linear*:
fixes $p :: 'a :: comm\text{-}semiring\text{-}1\ poly$
shows $coeff\ (pcompose\ p\ [:0,\ c:])\ i = c\ \hat{}\ i * coeff\ p\ i$
⟨*proof*⟩

lemma *coeff-pCons'*: $poly.coeff\ (pCons\ c\ p)\ n = (if\ n = 0\ then\ c\ else\ poly.coeff\ p\ (n - 1))$
⟨*proof*⟩

lemma *prod-smult*: $(\prod x{\in}A.\ Polynomial.smult\ (c\ x)\ (p\ x)) = Polynomial.smult\ (prod\ c\ A)\ (prod\ p\ A)$
⟨*proof*⟩

lemma *degree-higher-pderiv*: $Polynomial.degree\ ((pderiv\ \hat{}\hat{}\ n)\ p) = Polynomial.degree\ p - n$
for $p :: 'a{::}\{comm\text{-}semiring\text{-}1, semiring\text{-}no\text{-}zero\text{-}divisors, semiring\text{-}char\text{-}0\}\ poly$
⟨*proof*⟩

lemma *sum-to-poly*: $(\sum x{\in}A.\ [:f\ x:]) = [:\sum x{\in}A.\ f\ x:]$
⟨*proof*⟩

lemma *diff-to-poly*: $[:c:] - [:d:] = [:c - d:]$
⟨*proof*⟩

lemma *mult-to-poly*: $[:c:] * [:d:] = [:c * d:]$
⟨*proof*⟩

lemma *prod-to-poly*: $(\prod x{\in}A.\ [:f\ x:]) = [:\prod x{\in}A.\ f\ x:]$
⟨*proof*⟩

lemma *coeff-mult-0*: *poly.coeff* $(p * q)\ 0 =$ *poly.coeff* $p\ 0 *$ *poly.coeff* $q\ 0$
⟨*proof*⟩

lemma *card-poly-roots-bound*:
fixes $p :: 'a{::}\{$*comm-ring-1*,*ring-no-zero-divisors*$\}$ *poly*
assumes $p \neq 0$
shows *card* $\{x.$ *poly* $p\ x = 0\} \leq$ *degree* p
⟨*proof*⟩

lemma *poly-eqI-degree*:
fixes $p\ q :: 'a :: \{$*comm-ring-1*, *ring-no-zero-divisors*$\}$ *poly*
assumes $\bigwedge x.\ x \in A \Longrightarrow$ *poly* $p\ x =$ *poly* $q\ x$
assumes *card* $A >$ *degree* p *card* $A >$ *degree* q
shows $p = q$
⟨*proof*⟩

lemma *poly-root-order-induct* [*case-names 0 no-roots root*]:
fixes $p :: 'a ::$ *idom poly*
assumes $P\ 0\ \bigwedge p.\ (\bigwedge x.$ *poly* $p\ x \neq 0) \Longrightarrow P\ p$
$\bigwedge p\ x\ n.\ n > 0 \Longrightarrow$ *poly* $p\ x \neq 0 \Longrightarrow P\ p \Longrightarrow P\ ([:-x,\ 1:]\ \hat{}\ n * p)$
shows $P\ p$
⟨*proof*⟩

lemma *complex-poly-decompose*:
smult (*lead-coeff* p) $(\prod z|$*poly* $p\ z = 0.\ [:-z,\ 1:]\ \hat{}$ *order* $z\ p) = (p ::$ *complex poly*)
⟨*proof*⟩

lemma *order-pos-iff*: $p \neq 0 \Longrightarrow$ *order* $a\ p > 0 \longleftrightarrow$ *poly* $p\ a = 0$
⟨*proof*⟩

lift-definition *poly-roots-mset* :: $('a ::$ *idom*$)$ *poly* $\Rightarrow 'a$ *multiset* **is**
$\lambda p\ x.$ *if* $p = 0$ *then* 0 *else Polynomial.order* $x\ p$
⟨*proof*⟩

lemma *poly-roots-mset-0* [*simp*]: *poly-roots-mset* $0 = \{\#\}$

⟨*proof*⟩

lemma *count-poly-roots-mset* [*simp*]:
$p \neq 0 \Longrightarrow$ *count* (*poly-roots-mset p*) *a* = *order a p*
⟨*proof*⟩

lemma *set-count-poly-roots-mset* [*simp*]:
$p \neq 0 \Longrightarrow$ *set-mset* (*poly-roots-mset p*) = {*x*. *poly p x* = 0}
⟨*proof*⟩

lemma *image-prod-mset-multiplicity*:
prod-mset (*image-mset f M*) = *prod* (λ*x*. *f x* ^ *count M x*) (*set-mset M*)
⟨*proof*⟩

lemma *complex-poly-decompose-multiset*:
smult (*lead-coeff p*) ($\prod$ *x*∈#*poly-roots-mset p*. [:−*x*, 1:]) = (*p* :: *complex poly*)
⟨*proof*⟩

lemma (**in** *monoid-add*) *prod-list-prod-nth*:
prod-list xs = ($\prod$ *i*=0..<*length xs*. *xs* ! *i*)
⟨*proof*⟩

lemma *prod-zero-iff'*: *finite A* $\Longrightarrow$ *prod f A* = 0 ⟷ (∃ *x*∈*A*. *f x* = 0)
for *f* :: 'a ⇒ 'b :: {*comm-semiring-1*, *semiring-no-zero-divisors*}
⟨*proof*⟩

lemma *degree-prod-eq*: (⋀*x*. *x* ∈ *A* $\Longrightarrow$ *f x* ≠ 0) $\Longrightarrow$ *degree* (*prod f A*) = ($\sum$ *x*∈*A*. *degree* (*f x*))
for *f* :: 'a ⇒ 'b::{*comm-semiring-1*, *semiring-no-zero-divisors*} *poly*
⟨*proof*⟩

lemma *lead-coeff-prod*: (⋀*x*. *x* ∈ *A* $\Longrightarrow$ *f x* ≠ 0) $\Longrightarrow$ *lead-coeff* (*prod f A*) = ($\prod$ *x*∈*A*. *lead-coeff* (*f x*))
for *f* :: 'a ⇒ 'b :: {*comm-semiring-1*, *semiring-no-zero-divisors*} *poly*
⟨*proof*⟩

lemma *complex-poly-decompose'*:
obtains *root* **where** *smult* (*lead-coeff p*) ($\prod$ *i*<*degree p*. [:−*root i*, 1:]) = (*p* :: *complex poly*)
⟨*proof*⟩

lemma *rsquarefree-root-order*:
assumes *rsquarefree p poly p z* = 0 *p* ≠ 0
shows *order z p* = 1
⟨*proof*⟩

lemma *complex-poly-decompose-rsquarefree*:

assumes *rsquarefree p*
shows *smult* (*lead-coeff p*) ($\prod z | poly\ p\ z = 0.\ [:-z,\ 1:]$) = (*p* :: *complex poly*)
⟨*proof*⟩

lemma *pcompose-conjugates-integer*:
assumes $\bigwedge i.$ *poly.coeff p i* ∈ ℤ
shows *poly.coeff* (*pcompose p* [:0, i:] ∗ *pcompose p* [:0, −i:]) *i* ∈ ℤ
⟨*proof*⟩

lemma *algebraic-times-i*:
assumes *algebraic x*
shows *algebraic* (i ∗ *x*) *algebraic* (−i ∗ *x*)
⟨*proof*⟩

lemma *algebraic-times-i-iff*: *algebraic* (i ∗ *x*) ⟷ *algebraic x*
⟨*proof*⟩

lemma *ratpolyE*:
assumes ∀ *i*. *poly.coeff p i* ∈ ℚ
obtains *q* **where** *p* = *map-poly of-rat q*
⟨*proof*⟩

end

2 The Transcendence of π

theory *Pi-Transcendental*
imports
E-Transcendental.E-Transcendental
Symmetric-Polynomials.Symmetric-Polynomials
HOL−Real-Asymp.Real-Asymp
Pi-Transcendental-Polynomial-Library
begin

lemma *ring-homomorphism-to-poly* [*intro*]: *ring-homomorphism* ($\lambda i.$ [:*i*:])
⟨*proof*⟩

lemma (**in** *ring-closed*) *coeff-power-closed*:
($\bigwedge m.$ *coeff p m* ∈ *A*) ⟹ *coeff* (*p* ^ *n*) *m* ∈ *A*
⟨*proof*⟩

lemma (**in** *ring-closed*) *coeff-prod-closed*:
($\bigwedge x\ m.\ x \in X \Longrightarrow$ *coeff* (*f x*) *m* ∈ *A*) ⟹ *coeff* (*prod f X*) *m* ∈ *A*
⟨*proof*⟩

lemma *map-of-rat-of-int-poly* [*simp*]: *map-poly of-rat* (*of-int-poly p*) = *of-int-poly p*
⟨*proof*⟩

Given a polynomial with rational coefficients, we can obtain an integer polynomial that differs from it only by a nonzero constant by clearing the denominators.

lemma *ratpoly-to-intpoly*:
assumes $\forall i.$ *poly.coeff p i* $\in \mathbb{Q}$
obtains *q c* **where** $c \neq 0$ $p =$ *Polynomial.smult* (*inverse* (*of-nat c*)) (*of-int-poly q*)
⟨*proof*⟩

lemma *symmetric-mpoly-symmetric-sum*:
assumes $\bigwedge\pi.$ π *permutes A* $\Longrightarrow$ *g* π *permutes X*
assumes $\bigwedge x\ \pi.$ $x \in X \Longrightarrow$ π *permutes A* $\Longrightarrow$ *mpoly-map-vars* π (*f x*) = *f* (*g* π *x*)
shows *symmetric-mpoly A* ($\sum x{\in}X.$ *f x*)
⟨*proof*⟩

lemma *symmetric-mpoly-symmetric-prod*:
assumes *g permutes X*
assumes $\bigwedge x\ \pi.$ $x \in X \Longrightarrow$ π *permutes A* $\Longrightarrow$ *mpoly-map-vars* π (*f x*) = *f* (*g x*)
shows *symmetric-mpoly A* ($\prod x{\in}X.$ *f x*)
⟨*proof*⟩

We now prove the transcendence of $i\pi$, from which the transcendence of π will follow as a trivial corollary. The first proof of this was given by von Lindemann [4]. The central ingredient is the fundamental theorem of symmetric functions.

The proof can, by now, be considered folklore and one can easily find many similar variants of it, but we mostly follows the nice exposition given by Niven [3].

An independent previous formalisation in Coq that uses the same basic techniques was given by Bernard et al. [2]. They later also formalised the much stronger Lindemann–Weierstra theorem [1].

lemma *transcendental-i-pi*: ¬*algebraic* (i ∗ *pi*)
⟨*proof*⟩

theorem *transcendental-pi*: ¬*algebraic pi*
⟨*proof*⟩

end

References

[1] S. Bernard. Formalization of the Lindemann-Weierstrass Theorem. In *Interactive Theorem Proving*, Brasilia, Brazil, Sept. 2017.

[2] S. Bernard, Y. Bertot, L. Rideau, and P.-Y. Strub. Formal proofs of transcendence for e and pi as an application of multivariate and symmetric polynomials. In *Proceedings of the 5th ACM SIGPLAN Conference on Certified Programs and Proofs*, CPP 2016, pages 76–87, New York, NY, USA, 2016. ACM.

[3] I. Niven. The transcendence of π. *The American Mathematical Monthly*, 46(8):469–471, 1939.

[4] F. von Lindemann. Ueber die Zahl π. *Mathematische Annalen*, 20(2):213–225, Jun 1882.