Wieferich–Kempner Theorem

Jamie Chen

March 17, 2025

Abstract

This document presents a formalised proof of the Wieferich–Kempner Theorem, stating that all nonnegative integers can be expressed as the sum of nine nonnegative cubes. The source of the proof is the book "Additive Number Theory: The Classical Bases" by Melvyn B. Nathanson [2].

Contents

1 Technical Lemmas **3**

1.1 Lemma 2.1 in [2] 3

1.2 Lemma 2.2 in [2] 3

1.3 Lemma 2.3 in [2] 4

1.4 Lemma 2.4 in [2] 4

2 Wieferich–Kempner Theorem **5**

theory *Wieferich-Kempner*
imports
HOL−Number-Theory.Cong
HOL.Real
HOL.NthRoot
HOL.Transcendental
HOL−Library.Code-Target-Nat
Three-Squares.Three-Squares
Main
begin

fun *sumpow* :: *nat* ⇒ *nat list* ⇒ *nat* **where** [*code-computation-unfold*, *coercion-enabled*]:
sumpow p l = *fold* (+) (*map* (λ*x*. *x*^*p*) *l*) *0*
declare *sumpow.simps*[*code*]

definition *is-sumpow* :: *nat* ⇒ *nat* ⇒ *nat* ⇒ *bool*
where *is-sumpow p n m* ≡ ∃ *l*. *length l* = *n* ∧ *m* = *sumpow p l*

1 Technical Lemmas

We show four lemmas used in the main theorem.

1.1 Lemma 2.1 in [2]

lemma *sum-of-6-cubes*:
fixes *A m* :: *nat*
assumes *mLessASq*: $m \leq A^2$
assumes *mIsSum3Sq*: *is-sumpow 2 3 m*
shows *is-sumpow 3 6* $(6 * A * (A^2 + m))$
⟨*proof*⟩

1.2 Lemma 2.2 in [2]

lemma *if-cube-cong-then-cong*:
fixes *t* :: *nat*
fixes b_1 b_2 :: *int*
assumes *odd*: *odd* b_1 ∧ *odd* b_2
assumes $b_1 > 0 \wedge b_2 > 0$
assumes *tGeq1*: $t \geq 1$
assumes $[b_1\hat{}3 = b_2\hat{}3]$ (*mod* $2\hat{}t$)
shows $[b_1 \neq b_2]$ (*mod* $2\hat{}t$) ⟹ $[b_1\hat{}3 \neq b_2\hat{}3]$ (*mod* $2\hat{}t$)
⟨*proof*⟩

lemma *every-odd-nat-cong-cube*:
fixes *t w* :: *nat*
assumes *tPositive*: $t \geq 1$

assumes *wOdd*: *odd w*
shows $\exists\, b.\ odd\ b \wedge [w = b\hat{\ }3]\ (mod\ 2\hat{\ }t)$
⟨*proof*⟩

1.3 Lemma 2.3 in [2]

It is this section in which we use the Three Squares Theorem AFP Entry [1].

lemma *sum-of-3-squares-exceptions*:
fixes *m*::*nat*
assumes *notSum3Sq*: ¬*is-sumpow 2 3 m*
shows $6*m\ mod\ 96 \in \{0,72,42,90\}$
⟨*proof*⟩

lemma *values-geq-22-cubed-can-be-normalised*:
fixes *r* :: *nat*
assumes *rLarge*: $r \geq 10648$
obtains *d m* **where** $d \geq 0$ **and** $d \leq 22$ **and** $r = d\hat{\ }3 + 6*m$ **and** *is-sumpow 2 3 m*
⟨*proof*⟩

1.4 Lemma 2.4 in [2]

partial-function(*tailrec*) *list-builder* :: *nat* ⇒ *nat* ⇒ *nat list* ⇒ *nat list*
where [*code-computation-unfold*, *coercion-enabled*]:
list-builder m n l = (*if n* = *0 then l else* (*list-builder m* (*n*−*1*) (*m*#*l*)))
declare *list-builder*.*simps*[*code*]

partial-function(*tailrec*) *dec-list* :: *nat* ⇒ *nat list* ⇒ *nat list*
where [*code-computation-unfold*, *coercion-enabled*]:
dec-list depth l = (*if* (*tl l*) = [] *then list-builder* (*hd l* + *1*) (*depth*+*1*) [] *else*
if hd l ≤ *hd* (*tl l*) + *1 then dec-list* (*depth*+*1*) ((*hd* (*tl l*) + *1*)#(*tl* (*tl l*))) *else*
list-builder (*hd* (*tl l*) + *1*) (*depth* + *2*) (*tl* (*tl l*)))
declare *dec-list*.*simps*[*code*]

partial-function(*tailrec*) *sumcubepow-finder* ::*nat* ⇒ *nat list* ⇒ *nat list*
where [*code-computation-unfold*, *coercion-enabled*]:
sumcubepow-finder n l = (*if* (*sumpow 3 l* < *n*) *then*
(*sumcubepow-finder n* ((*Suc* (*hd l*))#(*tl l*)))
else if (*sumpow 3 l*) = *n then l else sumcubepow-finder n* (*dec-list 0 l*))
declare *sumcubepow-finder*.*simps*[*code*]

lemma *leq-40000-is-sum-of-9-cubes*:
fixes *n* :: *nat*
assumes $n \leq 40000$
shows *is-sumpow 3 9 n* **and** $n > 8042 \longrightarrow$ *is-sumpow 3 6 n*
⟨*proof*⟩

2 Wieferich–Kempner Theorem

Theorem 2.1 in [2]

theorem *Wieferich-Kempner*:
 fixes *N* :: *nat*
 shows *is-sumpow 3 9 N*
⟨*proof*⟩
end

References

[1] A. Danilkin and L. Chevalier. Three squares theorem. *Archive of Formal Proofs*, May 2023. https://isa-afp.org/entries/Three_Squares.html, Formal proof development.

[2] M. B. Nathanson. *Additive Number Theory: The Classical Bases*, volume 164 of *Graduate Texts in Mathematics*. Springer, New York, 1996.