Gauss Sums and the Pólya–Vinogradov Inequality

Rodrigo Raya and Manuel Eberl

March 17, 2025

Abstract

This article provides a full formalisation of Chapter 8 of Apostol's *Introduction to Analytic Number Theory* [1]. Subjects that are covered are:

- periodic arithmetic functions and their finite Fourier series
- (generalised) Ramanujan sums
- Gauss sums and separable characters
- induced moduli and primitive characters
- the Pólya–Vinogradov inequality

Contents

1 Auxiliary material

theory *Gauss-Sums-Auxiliary*
imports
Dirichlet-L.Dirichlet-Characters
Dirichlet-Series.Moebius-Mu
Dirichlet-Series.More-Totient
begin

1.1 Various facts

lemma *sum-div-reduce*:
fixes d :: *nat* **and** f :: *nat* ⇒ *complex*
assumes d *dvd* k $d > 0$
shows $(\sum n \mid n \in \{1..k\} \wedge d \textit{ dvd } n.\ f\ n) = (\sum c \in \{1..k \textit{ div } d\}.\ f\ (c*d))$
⟨*proof*⟩

lemma *prod-div-sub*:
fixes f :: *nat* ⇒ *complex*
assumes *finite* A $B \subseteq A$ $\forall b \in B.\ f\ b \neq 0$
shows $(\prod i \in A - B.\ f\ i) = ((\prod i \in A.\ f\ i) \textit{ div } (\prod i \in B.\ f\ i))$
⟨*proof*⟩

lemma *linear-gcd*:
fixes a b c d :: *nat*
assumes $a > 0$ $b > 0$ $c > 0$ $d > 0$
assumes *coprime* a c *coprime* b d
shows *gcd* $(a*b)$ $(c*d)$ = (*gcd* a d) $*$ (*gcd* b c)
⟨*proof*⟩

lemma *reindex-product-bij*:
fixes a b m k :: *nat*
defines $S \equiv \{(d1,d2).\ d1 \textit{ dvd gcd } a\ m \wedge d2 \textit{ dvd gcd } k\ b\}$
defines $T \equiv \{d.\ d \textit{ dvd } (\textit{gcd } a\ m) * (\textit{gcd } k\ b)\}$
defines $f \equiv (\lambda(d1,d2).\ d1 * d2)$
assumes *coprime* a k
shows *bij-betw* f S T
⟨*proof*⟩

lemma *p-div-set*:
shows $\{p.\ p \in \textit{prime-factors } a \wedge \neg\ p \textit{ dvd } N\}$ =
$(\{p.\ p \in \textit{prime-factors } (a*N)\} - \{p.\ p \in \textit{prime-factors } N\})$
(**is** *?A* = *?B*)
⟨*proof*⟩

lemma *coprime-iff-prime-factors-disjoint*:
fixes x y :: *'a* :: *factorial-semiring*
assumes $x \neq 0$ $y \neq 0$
shows *coprime* x y ⟷ *prime-factors* x ∩ *prime-factors* y = {}
⟨*proof*⟩

lemma *coprime-cong-prime-factors*:
fixes $x\ y :: 'a :: factorial\text{-}semiring\text{-}gcd$
assumes $x \neq 0\ y \neq 0\ x' \neq 0\ y' \neq 0$
assumes *prime-factors* $x =$ *prime-factors* x'
assumes *prime-factors* $y =$ *prime-factors* y'
shows *coprime* $x\ y \longleftrightarrow$ *coprime* $x'\ y'$
⟨*proof*⟩

lemma *moebius-prod-not-coprime*:
assumes $\neg$ *coprime* $N\ d$
shows *moebius-mu* $(N*d) = 0$
⟨*proof*⟩

Theorem 2.18

lemma *sum-divisors-moebius-mu-times-multiplicative*:
fixes $f :: nat \Rightarrow 'a :: \{comm\text{-}ring\text{-}1\}$
assumes *multiplicative-function* f **and** $n > 0$
shows $(\sum d \mid d\ dvd\ n.\ moebius\text{-}mu\ d * f\ d) = (\prod p{\in}prime\text{-}factors\ n.\ 1 - f\ p)$
⟨*proof*⟩

lemma *multiplicative-ind-coprime* [*intro*]: *multiplicative-function* (*ind* (*coprime* N))
⟨*proof*⟩

lemma *sum-divisors-moebius-mu-times-multiplicative-revisited*:
fixes $f :: nat \Rightarrow 'a :: \{comm\text{-}ring\text{-}1\}$
assumes *multiplicative-function* $f\ n > 0\ N > 0$
shows $(\sum d \mid d\ dvd\ n \wedge coprime\ N\ d.\ moebius\text{-}mu\ d * f\ d) =$
$(\prod p{\in}\{p.\ p \in prime\text{-}factors\ n \wedge \neg\ (p\ dvd\ N)\}.\ 1 - f\ p)$
⟨*proof*⟩

1.2 Neutral element of the Dirichlet product

definition *dirichlet-prod-neutral* $n =$ (*if* $n = 1$ *then* 1 *else* 0) **for** $n :: nat$

lemma *dirichlet-prod-neutral-intro*:
fixes $S :: nat \Rightarrow complex$ **and** $f :: nat \Rightarrow nat \Rightarrow complex$
defines $S \equiv (\lambda(n{::}nat).\ (\sum k \mid k \in \{1..n\} \wedge coprime\ k\ n.\ (f\ k\ n)))$
shows $S(n) = (\sum k \in \{1..n\}.\ f\ k\ n * dirichlet\text{-}prod\text{-}neutral\ (gcd\ k\ n))$
⟨*proof*⟩

lemma *dirichlet-prod-neutral-right-neutral*:
dirichlet-prod f *dirichlet-prod-neutral* $n = f\ n$ **if** $n > 0$ **for** $f :: nat \Rightarrow complex$ **and** n
⟨*proof*⟩

lemma *dirichlet-prod-neutral-left-neutral*:
dirichlet-prod dirichlet-prod-neutral $f\ n = f\ n$
if $n > 0$ **for** $f :: nat \Rightarrow complex$ **and** n

⟨*proof*⟩

corollary *I-right-neutral-0*:
fixes *f* :: *nat* ⇒ *complex*
assumes *f 0* = *0*
shows *dirichlet-prod f dirichlet-prod-neutral n* = *f n*
⟨*proof*⟩

1.3 Multiplicative functions

lemma *mult-id*: *multiplicative-function id*
⟨*proof*⟩

lemma *mult-moebius*: *multiplicative-function moebius-mu*
⟨*proof*⟩

lemma *mult-of-nat*: *multiplicative-function of-nat*
⟨*proof*⟩

lemma *mult-of-nat-c*: *completely-multiplicative-function of-nat*
⟨*proof*⟩

lemma *completely-multiplicative-nonzero*:
fixes *f* :: *nat* ⇒ *complex*
assumes *completely-multiplicative-function f*
d ≠ *0*
⋀*p*. *prime p* ⟹ *f*(*p*) ≠ *0*
shows *f*(*d*) ≠ *0*
⟨*proof*⟩

lemma *multipl-div*:
fixes *m k d1 d2* :: *nat* **and** *f* :: *nat* ⇒ *complex*
assumes *multiplicative-function f d1 dvd m d2 dvd k coprime m k*
shows *f* ((*m*∗*k*) *div* (*d1*∗*d2*)) = *f*(*m div d1*) ∗ *f*(*k div d2*)
⟨*proof*⟩

lemma *multipl-div-mono*:
fixes *m k d* :: *nat* **and** *f* :: *nat* ⇒ *complex*
assumes *completely-multiplicative-function f*
d dvd k d > *0*
⋀*p*. *prime p* ⟹ *f*(*p*) ≠ *0*
shows *f* (*k div d*) = *f*(*k*) *div f*(*d*)
⟨*proof*⟩

lemma *comp-to-mult*: *completely-multiplicative-function f* ⟹
multiplicative-function f
⟨*proof*⟩

end

2 Periodic arithmetic functions

theory *Periodic-Arithmetic*
imports
Complex-Main
HOL−Number-Theory.Cong
begin

definition
periodic-arithmetic f k = ($\forall$ *n*. *f* (*n*+*k*) = *f n*)
for *n* :: *int* **and** *k* :: *nat* **and** *f* :: *nat* ⇒ *complex*

lemma *const-periodic-arithmetic*: *periodic-arithmetic* (λ*x*. *y*) *k*
⟨*proof*⟩

lemma *add-periodic-arithmetic*:
fixes *f g* :: *nat* ⇒ *complex*
assumes *periodic-arithmetic f k*
assumes *periodic-arithmetic g k*
shows *periodic-arithmetic* (λ*n*. *f n* + *g n*) *k*
⟨*proof*⟩

lemma *mult-periodic-arithmetic*:
fixes *f g* :: *nat* ⇒ *complex*
assumes *periodic-arithmetic f k*
assumes *periodic-arithmetic g k*
shows *periodic-arithmetic* (λ*n*. *f n* ∗ *g n*) *k*
⟨*proof*⟩

lemma *scalar-mult-periodic-arithmetic*:
fixes *f* :: *nat* ⇒ *complex* **and** *a* :: *complex*
assumes *periodic-arithmetic f k*
shows *periodic-arithmetic* (λ*n*. *a* ∗ *f n*) *k*
⟨*proof*⟩

lemma *fin-sum-periodic-arithmetic-set*:
fixes *f g* :: *nat* ⇒ *complex*
assumes ∀ *i*∈*A*. *periodic-arithmetic* (*h i*) *k*
shows *periodic-arithmetic* (λ*n*. $\sum$ *i* ∈ *A*. *h i n*) *k*
⟨*proof*⟩

lemma *mult-period*:
assumes *periodic-arithmetic g k*
shows *periodic-arithmetic g* (*k*∗*q*)
⟨*proof*⟩

lemma *unique-periodic-arithmetic-extension*:
assumes *k* > *0*
assumes ∀ *j*<*k*. *g j* = *h j*

assumes *periodic-arithmetic g k* **and** *periodic-arithmetic h k*
shows $g\ i = h\ i$
⟨*proof*⟩

lemma *periodic-arithmetic-sum-periodic-arithmetic*:
assumes *periodic-arithmetic f k*
shows $(\sum l \in \{m..n\}.\ f\ l) = (\sum l \in \{m+k..n+k\}.\ f\ l)$
⟨*proof*⟩

lemma *mod-periodic-arithmetic*:
fixes *n m* :: *nat*
assumes *periodic-arithmetic f k*
assumes $n\ mod\ k = m\ mod\ k$
shows $f\ n = f\ m$
⟨*proof*⟩

lemma *cong-periodic-arithmetic*:
assumes *periodic-arithmetic f k* $[a = b]\ (mod\ k)$
shows $f\ a = f\ b$
⟨*proof*⟩

lemma *cong-nat-imp-eq*:
fixes *m* :: *nat*
assumes $m > 0$ $x \in \{a..<a+m\}$ $y \in \{a..<a+m\}$ $[x = y]\ (mod\ m)$
shows $x = y$
⟨*proof*⟩

lemma *inj-on-mod-nat*:
fixes *m* :: *nat*
assumes $m > 0$
shows *inj-on* $(\lambda x.\ x\ mod\ m)\ \{a..<a+m\}$
⟨*proof*⟩

lemma *bij-betw-mod-nat-atLeastLessThan*:
fixes *k d* :: *nat*
assumes $k > 0$
defines $g \equiv (\lambda i.\ nat\ ((int\ i - int\ d)\ mod\ int\ k) + d)$
shows *bij-betw* $(\lambda i.\ i\ mod\ k)\ \{d..<d+k\}\ \{..<k\}$
⟨*proof*⟩

lemma *periodic-arithmetic-sum-periodic-arithmetic-shift*:
fixes *k d* :: *nat*
assumes *periodic-arithmetic f k* $k > 0$ $d > 0$
shows $(\sum l \in \{0..k-1\}.\ f\ l) = (\sum l \in \{d..d+k-1\}.\ f\ l)$
⟨*proof*⟩

lemma *self-bij-0-k*:
fixes *a k* :: *nat*
assumes *coprime a k* $[a*i = 1]\ (mod\ k)$ $k > 0$

shows *bij-betw* ($\lambda r.\ r*a\ mod\ k$) {*0..k−1*} {*0..k−1*}
⟨*proof*⟩

lemma *periodic-arithmetic-homothecy*:
assumes *periodic-arithmetic f k*
shows *periodic-arithmetic* ($\lambda l.\ f\ (l*a)$) *k*
⟨*proof*⟩

theorem *periodic-arithmetic-remove-homothecy*:
assumes *coprime a k periodic-arithmetic f k k* $>$ *0*
shows ($\sum l=1..k.\ f\ l$) = ($\sum l=1..k.\ f\ (l*a)$)
⟨*proof*⟩

end

theory *Complex-Roots-Of-Unity*
imports
HOL−Analysis.Analysis
Periodic-Arithmetic
begin

3 Complex roots of unity

definition
unity-root k n = *cis* (*2* ∗ *pi* ∗ *of-int n* / *of-nat k*)

lemma
unity-root-k-0 [*simp*]: *unity-root k 0* = *1* **and**
unity-root-0-n [*simp*]: *unity-root 0 n* = *1*
⟨*proof*⟩

lemma *unity-root-conv-exp*:
unity-root k n = *exp* (*of-real* (*2*∗*pi*∗*n*/*k*) ∗ i)
⟨*proof*⟩

lemma *unity-root-mod*:
unity-root k (*n mod int k*) = *unity-root k n*
⟨*proof*⟩

lemma *unity-root-cong*:
assumes [*m* = *n*] (*mod int k*)
shows *unity-root k m* = *unity-root k n*
⟨*proof*⟩

lemma *unity-root-mod-nat*:
unity-root k (*nat* (*n mod int k*)) = *unity-root k n*
⟨*proof*⟩

lemma *unity-root-eqD*:

assumes *gr*: $k > 0$
assumes *eq*: *unity-root k i* = *unity-root k j*
shows *i mod k* = *j mod k*
⟨*proof*⟩

lemma *unity-root-eq-1-iff*:
fixes *k n* :: *nat*
assumes $k > 0$
shows *unity-root k n* = *1* ⟷ *k dvd n*
⟨*proof*⟩

lemma *unity-root-pow*: *unity-root k n* ^ *m* = *unity-root k* (*n* ∗ *m*)
⟨*proof*⟩

lemma *unity-root-add*: *unity-root k* (*m* + *n*) = *unity-root k m* ∗ *unity-root k n*
⟨*proof*⟩

lemma *unity-root-uminus*: *unity-root k* (−*m*) = *cnj* (*unity-root k m*)
⟨*proof*⟩

lemma *inverse-unity-root*: *inverse* (*unity-root k m*) = *cnj* (*unity-root k m*)
⟨*proof*⟩

lemma *unity-root-diff*: *unity-root k* (*m* − *n*) = *unity-root k m* ∗ *cnj* (*unity-root k n*)
⟨*proof*⟩

lemma *unity-root-eq-1-iff-int*:
fixes *k* :: *nat* **and** *n* :: *int*
assumes $k > 0$
shows *unity-root k n* = *1* ⟷ *k dvd n*
⟨*proof*⟩

lemma *unity-root-eq-1* [*simp*]: *int k dvd n* ⟹ *unity-root k n* = *1*
⟨*proof*⟩

lemma *unity-periodic-arithmetic*:
periodic-arithmetic (*unity-root k*) *k*
⟨*proof*⟩

lemma *unity-periodic-arithmetic-mult*:
periodic-arithmetic (λ*n*. *unity-root k* (*m* ∗ *int n*)) *k*
⟨*proof*⟩

lemma *unity-root-periodic-arithmetic-mult-minus*:
shows *periodic-arithmetic* (λ*i*. *unity-root k* (−*int i*∗*int m*)) *k*
⟨*proof*⟩

lemma *unity-div*:

fixes *a* :: *int* **and** *d* :: *nat*
assumes *d dvd k*
shows *unity-root k* (*a*∗*d*) = *unity-root* (*k div d*) *a*
⟨*proof*⟩

lemma *unity-div-num*:
assumes *k* > *0* *d* > *0* *d dvd k*
shows *unity-root k* (*x* ∗ (*k div d*)) = *unity-root d x*
⟨*proof*⟩

4 Geometric sums of roots of unity

Apostol calls these 'geometric sums', which is a bit too generic. We therefore decided to refer to them as 'sums of roots of unity'.

definition *unity-root-sum k n* = (∑ *m*<*k*. *unity-root k* (*n* ∗ *of-nat m*))

lemma *unity-root-sum-0-left* [*simp*]: *unity-root-sum 0 n* = *0* **and**
unity-root-sum-0-right [*simp*]: *k* > *0* ⟹ *unity-root-sum k 0* = *k*
⟨*proof*⟩

Theorem 8.1

theorem *unity-root-sum*:
fixes *k* :: *nat* **and** *n* :: *int*
assumes *gr*: *k* ≥ *1*
shows *k dvd n* ⟹ *unity-root-sum k n* = *k*
and ¬*k dvd n* ⟹ *unity-root-sum k n* = *0*
⟨*proof*⟩

corollary *unity-root-sum-periodic-arithmetic*:
periodic-arithmetic (*unity-root-sum k*) *k*
⟨*proof*⟩

lemma *unity-root-sum-nonzero-iff*:
fixes *r* :: *int*
assumes *k* ≥ *1* **and** *r* ∈ {−*k*<..<*k*}
shows *unity-root-sum k r* ≠ *0* ⟷ *r* = *0*
⟨*proof*⟩

end

5 Finite Fourier series

theory *Finite-Fourier-Series*
imports
Polynomial-Interpolation.Lagrange-Interpolation
Complex-Roots-Of-Unity
begin

5.1 Auxiliary facts

lemma *lagrange-exists*:
assumes *d*: *distinct* (*map fst zs-ws*)
defines *e*: (*p* :: *complex poly*) ≡ *lagrange-interpolation-poly zs-ws*
shows *degree p* ≤ (*length zs-ws*)−*1*
(∀ *x y*. (*x*,*y*) ∈ *set zs-ws* ⟶ *poly p x* = *y*)
⟨*proof*⟩

lemma *lagrange-unique*:
assumes *o*: *length zs-ws* > *0*
assumes *d*: *distinct* (*map fst zs-ws*)
assumes *1*: *degree* (*p1* :: *complex poly*) ≤ (*length zs-ws*)−*1* ∧
(∀ *x y*. (*x*,*y*) ∈ *set zs-ws* ⟶ *poly p1 x* = *y*)
assumes *2*: *degree* (*p2* :: *complex poly*) ≤ (*length zs-ws*)−*1* ∧
(∀ *x y*. (*x*,*y*) ∈ *set zs-ws* ⟶ *poly p2 x* = *y*)
shows *p1* = *p2*
⟨*proof*⟩

Theorem 8.2

corollary *lagrange*:
assumes *length zs-ws* > *0 distinct* (*map fst zs-ws*)
shows (∃! (*p* :: *complex poly*).
degree p ≤ *length zs-ws* − *1* ∧
(∀ *x y*. (*x*, *y*) ∈ *set zs-ws* ⟶ *poly p x* = *y*))
⟨*proof*⟩

lemma *poly-altdef′*:
assumes *gr*: *k* ≥ *degree p*
shows *poly p* (*z*::*complex*) = (∑ *i*≤*k*. *coeff p i* ∗ *z* ^ *i*)
⟨*proof*⟩

5.2 Definition and uniqueness

definition *finite-fourier-poly* :: *complex list* ⇒ *complex poly* **where**
finite-fourier-poly ws =
(*let k* = *length ws*
in poly-of-list [*1* / *k* ∗ (∑ *m*<*k*. *ws* ! *m* ∗ *unity-root k* (−*n*∗*m*)). *n* ← [*0*..<*k*]])

lemma *degree-poly-of-list-le*: *degree* (*poly-of-list ws*) ≤ *length ws* − *1*
⟨*proof*⟩

lemma *degree-finite-fourier-poly*: *degree* (*finite-fourier-poly ws*) ≤ *length ws* − *1*
⟨*proof*⟩

lemma *coeff-finite-fourier-poly*:
assumes *n* < *length ws*
defines *k* ≡ *length ws*
shows *coeff* (*finite-fourier-poly ws*) *n* =
(*1*/*k*) ∗ (∑ *m* < *k*. *ws* ! *m* ∗ *unity-root k* (−*n*∗*m*))

⟨*proof*⟩

lemma *poly-finite-fourier-poly*:
fixes m :: *int* **and** *ws*
defines $k \equiv$ *length ws*
assumes $m \in \{0..<k\}$
assumes $m <$ *length ws*
shows *poly* (*finite-fourier-poly ws*) (*unity-root k m*) = *ws* ! (*nat m*)
⟨*proof*⟩

Theorem 8.3

theorem *finite-fourier-poly-unique*:
assumes *length ws* > 0
defines $k \equiv$ *length ws*
assumes (*degree* $p \leq k - 1$)
assumes ($\forall m \leq k-1$. (*ws* ! *m*) = *poly p* (*unity-root k m*))
shows p = *finite-fourier-poly ws*
⟨*proof*⟩

The following alternative formulation returns a coefficient

definition *finite-fourier-poly'* :: (*nat* ⇒ *complex*) ⇒ *nat* ⇒ *complex poly* **where**
finite-fourier-poly' ws k =
(*poly-of-list* [1 / k ∗ ($\sum m<k$. (*ws m*) ∗ *unity-root k* ($-n*m$)). $n \leftarrow [0..<k]$])

lemma *finite-fourier-poly'-conv-finite-fourier-poly*:
finite-fourier-poly' ws k = *finite-fourier-poly* [*ws n*. $n \leftarrow [0..<k]$]
⟨*proof*⟩

lemma *coeff-finite-fourier-poly'*:
assumes $n < k$
shows *coeff* (*finite-fourier-poly' ws k*) n =
($1/k$) ∗ ($\sum m < k$. (*ws m*) ∗ *unity-root k* ($-n*m$))
⟨*proof*⟩

lemma *degree-finite-fourier-poly'*: *degree* (*finite-fourier-poly' ws k*) $\leq k - 1$
⟨*proof*⟩

lemma *poly-finite-fourier-poly'*:
fixes m :: *int* **and** k
assumes $m \in \{0..<k\}$
shows *poly* (*finite-fourier-poly' ws k*) (*unity-root k m*) = *ws* (*nat m*)
⟨*proof*⟩

lemma *finite-fourier-poly'-unique*:
assumes $k > 0$
assumes *degree* $p \leq k - 1$
assumes $\forall m \leq k-1$. *ws m* = *poly p* (*unity-root k m*)
shows p = *finite-fourier-poly' ws k*
⟨*proof*⟩

lemma *fourier-unity-root*:
fixes *k* :: *nat*
assumes *k > 0*
shows *poly* (*finite-fourier-poly′ f k*) (*unity-root k m*) =
($\sum$ *n<k.1/k∗*($\sum$ *m<k.*(*f m*)∗*unity-root k* (−*n∗m*))∗*unity-root k* (*m∗n*))
⟨*proof*⟩

5.3 Expansion of an arithmetical function

Theorem 8.4

theorem *fourier-expansion-periodic-arithmetic*:
assumes *k > 0*
assumes *periodic-arithmetic f k*
defines *g* ≡ (λ*n.* (*1* / *k*) ∗ ($\sum$ *m<k. f m* ∗ *unity-root k* (−*n* ∗ *m*)))
shows *periodic-arithmetic g k*
and *f m* = ($\sum$ *n<k. g n* ∗ *unity-root k* (*m* ∗ *n*))
⟨*proof*⟩

theorem *fourier-expansion-periodic-arithmetic-unique*:
fixes *f g* :: *nat* ⇒ *complex*
assumes *k > 0*
assumes *periodic-arithmetic f k* **and** *periodic-arithmetic g k*
assumes ⋀*m. m < k* ⟹ *f m* = ($\sum$ *n<k. g n* ∗ *unity-root k* (*int* (*m* ∗ *n*)))
shows *g n* = (*1* / *k*) ∗ ($\sum$ *m<k. f m* ∗ *unity-root k* (−*n* ∗ *m*))
⟨*proof*⟩

end

6 Ramanujan sums

theory *Ramanujan-Sums*
imports
Dirichlet-Series.Moebius-Mu
Gauss-Sums-Auxiliary
Finite-Fourier-Series
begin

6.1 Basic sums

definition *ramanujan-sum* :: *nat* ⇒ *nat* ⇒ *complex*
where *ramanujan-sum k n* = ($\sum$ *m* | *m* ∈ {*1..k*} ∧ *coprime m k. unity-root k* (*m∗n*))

notation *ramanujan-sum* (‹*c*›)

lemma *ramanujan-sum-0-n* [*simp*]: *c 0 n = 0*
⟨*proof*⟩

lemma *sum-coprime-conv-dirichlet-prod-moebius-mu*:
fixes $F\ S :: nat \Rightarrow complex$ **and** $f :: nat \Rightarrow nat \Rightarrow complex$
defines $F \equiv (\lambda n.\ (\sum k \in \{1..n\}.\ f\ k\ n))$
defines $S \equiv (\lambda n.\ (\sum k \mid k \in \{1..n\} \wedge coprime\ k\ n\ .\ f\ k\ n))$
assumes $\bigwedge a\ b\ d.\ d\ dvd\ a \Longrightarrow d\ dvd\ b \Longrightarrow f\ (a\ div\ d)\ (b\ div\ d) = f\ a\ b$
shows $S\ n = dirichlet\text{-}prod\ moebius\text{-}mu\ F\ n$
⟨*proof*⟩

lemma *dirichlet-prod-neutral-sum*:
$dirichlet\text{-}prod\text{-}neutral\ n = (\sum k = 1..n.\ unity\text{-}root\ n\ k)$ **for** $n :: nat$
⟨*proof*⟩

lemma *moebius-coprime-sum*:
$moebius\text{-}mu\ n = (\sum k \mid k \in \{1..n\} \wedge coprime\ k\ n\ .\ unity\text{-}root\ n\ (int\ k))$
⟨*proof*⟩

corollary *ramanujan-sum-1-right* [*simp*]: $c\ k\ (Suc\ 0) = moebius\text{-}mu\ k$
⟨*proof*⟩

lemma *ramanujan-sum-dvd-eq-totient*:
assumes $k\ dvd\ n$
shows $c\ k\ n = totient\ k$
⟨*proof*⟩

6.2 Generalised sums

definition *gen-ramanujan-sum* :: $(nat \Rightarrow complex) \Rightarrow (nat \Rightarrow complex) \Rightarrow nat \Rightarrow nat \Rightarrow complex$ **where**
$gen\text{-}ramanujan\text{-}sum\ f\ g = (\lambda k\ n.\ \sum d \mid d\ dvd\ gcd\ n\ k.\ f\ d * g\ (k\ div\ d))$

notation *gen-ramanujan-sum* (‹*s*›)

lemma *gen-ramanujan-sum-k-1*: $s\ f\ g\ k\ 1 = f\ 1 * g\ k$
⟨*proof*⟩

lemma *gen-ramanujan-sum-1-n*: $s\ f\ g\ 1\ n = f\ 1 * g\ 1$
⟨*proof*⟩

lemma *gen-ramanujan-sum-periodic*: $periodic\text{-}arithmetic\ (s\ f\ g\ k)\ k$
⟨*proof*⟩

Theorem 8.5

theorem *gen-ramanujan-sum-fourier-expansion*:
fixes $f\ g :: nat \Rightarrow complex$ **and** $a :: nat \Rightarrow nat \Rightarrow complex$
assumes $k > 0$
defines $a \equiv (\lambda k\ m.\ (1/k) * (\sum d \mid d\ dvd\ (gcd\ m\ k).\ g\ d * f\ (k\ div\ d) * d))$
shows $s\ f\ g\ k\ n = (\sum m \le k-1.\ a\ k\ m * unity\text{-}root\ k\ (m*n))$
⟨*proof*⟩

Theorem 8.6

theorem *ramanujan-sum-dirichlet-form*:
fixes *k n* :: *nat*
assumes *k* > *0*
shows *c k n* = ($\sum$ *d* | *d dvd gcd n k*. *d* ∗ *moebius-mu* (*k div d*))
⟨*proof*⟩

corollary *ramanujan-sum-conv-gen-ramanujan-sum*:
k > *0* ⟹ *c k n* = *s id moebius-mu k n*
⟨*proof*⟩

Theorem 8.7

theorem *gen-ramanujan-sum-distrib*:
fixes *f g* :: *nat* ⇒ *complex*
assumes *a* > *0 b* > *0 m* > *0 k* > *0*
assumes *coprime a k coprime b m coprime k m*
assumes *multiplicative-function f* **and**
multiplicative-function g
shows *s f g* (*m*∗*k*) (*a*∗*b*) = *s f g m a* ∗ *s f g k b*
⟨*proof*⟩

corollary *gen-ramanujan-sum-distrib-right*:
fixes *f g* :: *nat* ⇒ *complex*
assumes *a* > *0* **and** *b* > *0* **and** *m* > *0*
assumes *coprime b m*
assumes *multiplicative-function f* **and**
multiplicative-function g
shows *s f g m* (*a* ∗ *b*) = *s f g m a*
⟨*proof*⟩

corollary *gen-ramanujan-sum-distrib-left*:
fixes *f g* :: *nat* ⇒ *complex*
assumes *a* > *0* **and** *k* > *0* **and** *m* > *0*
assumes *coprime a k* **and** *coprime k m*
assumes *multiplicative-function f* **and**
multiplicative-function g
shows *s f g* (*m*∗*k*) *a* = *s f g m a* ∗ *g k*
⟨*proof*⟩

corollary *ramanujan-sum-distrib*:
assumes *a* > *0* **and** *k* > *0* **and** *m* > *0* **and** *b* > *0*
assumes *coprime a k coprime b m coprime m k*
shows *c* (*m*∗*k*) (*a*∗*b*) = *c m a* ∗ *c k b*
⟨*proof*⟩

corollary *ramanujan-sum-distrib-right*:
assumes *a* > *0* **and** *k* > *0* **and** *m* > *0* **and** *b* > *0*
assumes *coprime b m*
shows *c m* (*a*∗*b*) = *c m a*

⟨*proof*⟩

corollary *ramanujan-sum-distrib-left*:
assumes $a > 0$ $k > 0$ $m > 0$
assumes *coprime a k coprime m k*
shows $c\ (m*k)\ a = c\ m\ a * \textit{moebius-mu}\ k$
⟨*proof*⟩

lemma *dirichlet-prod-completely-multiplicative-left*:
fixes $f\ h :: nat \Rightarrow complex$ **and** $k :: nat$
defines $g \equiv (\lambda k.\ \textit{moebius-mu}\ k * h\ k)$
defines $F \equiv \textit{dirichlet-prod}\ f\ g$
assumes $k > 0$
assumes *completely-multiplicative-function f*
multiplicative-function h
assumes $\bigwedge p.\ prime\ p \Longrightarrow f(p) \neq 0 \wedge f(p) \neq h(p)$
shows $F\ k = f\ k * (\prod p{\in}\textit{prime-factors}\ k.\ 1 - h\ p\ /\ f\ p)$
⟨*proof*⟩

Theorem 8.8

theorem *gen-ramanujan-sum-dirichlet-expr*:
fixes $f\ h :: nat \Rightarrow complex$ **and** $n\ k :: nat$
defines $g \equiv (\lambda k.\ \textit{moebius-mu}\ k * h\ k)$
defines $F \equiv \textit{dirichlet-prod}\ f\ g$
defines $N \equiv k\ div\ gcd\ n\ k$
assumes *completely-multiplicative-function f*
multiplicative-function h
assumes $\bigwedge p.\ prime\ p \Longrightarrow f(p) \neq 0 \wedge f(p) \neq h(p)$
assumes $k > 0$ $n > 0$
shows $s\ f\ g\ k\ n = (F(k)*g(N))\ div\ (F(N))$
⟨*proof*⟩

lemma *totient-conv-moebius-mu-of-nat*:
of-nat (*totient n*) = *dirichlet-prod moebius-mu of-nat n*
⟨*proof*⟩

corollary *ramanujan-sum-k-n-dirichlet-expr*:
fixes $k\ n :: nat$
assumes $k > 0$ $n > 0$
shows $c\ k\ n = \textit{of-nat}\ (totient\ k) *$
moebius-mu (*k div gcd n k*) *div*
of-nat (*totient* (*k div gcd n k*))
⟨*proof*⟩

no-notation *ramanujan-sum* (‹*c*›)
no-notation *gen-ramanujan-sum* (‹*s*›)

end

theory *Gauss-Sums*
imports
HOL−Algebra.Coset
HOL−Real-Asymp.Real-Asymp
Ramanujan-Sums
begin

7 Gauss sums

bundle *vec-lambda-syntax*
begin
notation *vec-lambda* (**binder** ‹χ› *10*)
end

unbundle *no vec-lambda-syntax*

7.1 Definition and basic properties

context *dcharacter*
begin

lemma *dir-periodic-arithmetic*: *periodic-arithmetic* χ *n*
⟨*proof*⟩

definition *gauss-sum k* = (∑ *m* = *1..n* . χ(*m*) ∗ *unity-root n* (*m*∗*k*))

lemma *gauss-sum-periodic*:
periodic-arithmetic (λ*n*. *gauss-sum n*) *n*
⟨*proof*⟩

lemma *ramanujan-sum-conv-gauss-sum*:
assumes χ = *principal-dchar n*
shows *ramanujan-sum n k* = *gauss-sum k*
⟨*proof*⟩

lemma *cnj-mult-self*:
assumes *coprime k n*
shows *cnj* (χ *k*) ∗ χ *k* = *1*
⟨*proof*⟩

Theorem 8.9

theorem *gauss-sum-reduction*:
assumes *coprime k n*
shows *gauss-sum k* = *cnj* (χ *k*) ∗ *gauss-sum 1*
⟨*proof*⟩

The following variant takes an integer argument instead.

definition *gauss-sum-int k* = ($\sum$ *m=1..n. χ m ∗ unity-root n (int m∗k)*)

sublocale *gauss-sum-int*: *periodic-fun-simple gauss-sum-int int n*
⟨*proof*⟩

lemma *gauss-sum-int-cong*:
assumes [*a* = *b*] (*mod int n*)
shows *gauss-sum-int a* = *gauss-sum-int b*
⟨*proof*⟩

lemma *gauss-sum-conv-gauss-sum-int*:
gauss-sum k = *gauss-sum-int* (*int k*)
⟨*proof*⟩

lemma *gauss-sum-int-conv-gauss-sum*:
gauss-sum-int k = *gauss-sum* (*nat* (*k mod n*))
⟨*proof*⟩

lemma *gauss-int-periodic*: *periodic-arithmetic gauss-sum-int n*
⟨*proof*⟩

proposition *dcharacter-fourier-expansion*:
χ m = ($\sum$ *k=1..n. 1 / n ∗ gauss-sum-int* (−*k*) ∗ *unity-root n* (*m∗k*))
⟨*proof*⟩

7.2 Separability

definition *separable k* ⟷ *gauss-sum k* = *cnj* (*χ k*) ∗ *gauss-sum 1*

corollary *gauss-coprime-separable*:
assumes *coprime k n*
shows *separable k*
⟨*proof*⟩

Theorem 8.10

theorem *global-separability-condition*:
(∀ *n*>*0*. *separable n*) ⟷ (∀ *k*>*0*. ¬*coprime k n* ⟶ *gauss-sum k* = *0*)
⟨*proof*⟩

lemma *of-real-moebius-mu* [*simp*]: *of-real* (*moebius-mu k*) = *moebius-mu k*
⟨*proof*⟩

corollary *principal-not-totally-separable*:
assumes *χ* = *principal-dchar n*
shows ¬(∀ *k* > *0*. *separable k*)
⟨*proof*⟩

Theorem 8.11

theorem *gauss-sum-1-mod-square-eq-k*:

assumes $(\forall k.\ k > 0 \longrightarrow separable\ k)$
shows $norm\ (gauss\text{-}sum\ 1)\ \hat{}\ 2 = real\ n$
⟨*proof*⟩

Theorem 8.12

theorem *gauss-sum-nonzero-noncoprime-necessary-condition*:
assumes $gauss\text{-}sum\ k \neq 0\ \neg coprime\ k\ n\ k > 0$
defines $d \equiv n\ div\ gcd\ k\ n$
assumes $coprime\ a\ n\ [a = 1]\ (mod\ d)$
shows $d\ dvd\ n\ d < n\ \chi\ a = 1$
⟨*proof*⟩

7.3 Induced moduli and primitive characters

definition $induced\text{-}modulus\ d \longleftrightarrow d\ dvd\ n \wedge (\forall a.\ coprime\ a\ n \wedge [a = 1]\ (mod\ d) \longrightarrow \chi\ a = 1)$

lemma *induced-modulus-dvd*: $induced\text{-}modulus\ d \Longrightarrow d\ dvd\ n$
⟨*proof*⟩

lemma *induced-modulusI* [*intro?*]:
$d\ dvd\ n \Longrightarrow (\bigwedge a.\ coprime\ a\ n \Longrightarrow [a = 1]\ (mod\ d) \Longrightarrow \chi\ a = 1) \Longrightarrow induced\text{-}modulus\ d$
⟨*proof*⟩

lemma *induced-modulusD*: $induced\text{-}modulus\ d \Longrightarrow coprime\ a\ n \Longrightarrow [a = 1]\ (mod\ d) \Longrightarrow \chi\ a = 1$
⟨*proof*⟩

lemma *zero-not-ind-mod*: $\neg induced\text{-}modulus\ 0$
⟨*proof*⟩

lemma *div-gcd-dvd1*: $(a :: 'a :: semiring\text{-}gcd)\ div\ gcd\ a\ b\ dvd\ a$
⟨*proof*⟩

lemma *div-gcd-dvd2*: $(b :: 'a :: semiring\text{-}gcd)\ div\ gcd\ a\ b\ dvd\ b$
⟨*proof*⟩

lemma *g-non-zero-ind-mod*:
assumes $gauss\text{-}sum\ k \neq 0\ \neg coprime\ k\ n\ k > 0$
shows $induced\text{-}modulus\ (n\ div\ gcd\ k\ n)$
⟨*proof*⟩

lemma *induced-modulus-modulus*: $induced\text{-}modulus\ n$
⟨*proof*⟩

Theorem 8.13

theorem *one-induced-iff-principal*:
$induced\text{-}modulus\ 1 \longleftrightarrow \chi = principal\text{-}dchar\ n$

⟨*proof*⟩

end

locale *primitive-dchar* = *dcharacter* +
assumes *no-induced-modulus*: $\neg(\exists\, d{<}n.\ \textit{induced-modulus}\ d)$

locale *nonprimitive-dchar* = *dcharacter* +
assumes *induced-modulus*: $\exists\, d{<}n.\ \textit{induced-modulus}\ d$

lemma (**in** *nonprimitive-dchar*) *nonprimitive*: $\neg\textit{primitive-dchar}\ n\ \chi$
⟨*proof*⟩

lemma (**in** *dcharacter*) *primitive-dchar-iff*:
$\textit{primitive-dchar}\ n\ \chi \longleftrightarrow \neg(\exists\, d{<}n.\ \textit{induced-modulus}\ d)$
⟨*proof*⟩

lemma (**in** *residues-nat*) *principal-not-primitive*:
$\neg\textit{primitive-dchar}\ n\ (\textit{principal-dchar}\ n)$
⟨*proof*⟩

lemma (**in** *dcharacter*) *not-primitive-imp-nonprimitive*:
assumes $\neg\textit{primitive-dchar}\ n\ \chi$
shows *nonprimitive-dchar* $n\ \chi$
⟨*proof*⟩

Theorem 8.14

theorem (**in** *dcharacter*) *prime-nonprincipal-is-primitive*:
assumes *prime n*
assumes $\chi \neq \textit{principal-dchar}\ n$
shows *primitive-dchar* $n\ \chi$
⟨*proof*⟩

Theorem 8.15

corollary (**in** *primitive-dchar*) *primitive-encoding*:
$\forall\, k{>}0.\ \neg\textit{coprime}\ k\ n \longrightarrow \textit{gauss-sum}\ k = 0$
$\forall\, k{>}0.\ \textit{separable}\ k$
$\textit{norm}\ (\textit{gauss-sum}\ 1)\ \hat{}\ 2 = n$
⟨*proof*⟩

Theorem 8.16

lemma (**in** *dcharacter*) *induced-modulus-altdef1*:
$\textit{induced-modulus}\ d \longleftrightarrow$
$d\ \textit{dvd}\ n \wedge (\forall\, a\ b.\ \textit{coprime}\ a\ n \wedge \textit{coprime}\ b\ n \wedge [a = b]\ (\textit{mod}\ d) \longrightarrow \chi\ a = \chi\ b)$
⟨*proof*⟩

Exercise 8.4

lemma *induced-modulus-altdef2-lemma*:

fixes *n a d q* :: *nat*
defines $q \equiv (\prod\ p \mid prime\ p \wedge p\ dvd\ n \wedge \neg\ (p\ dvd\ a).\ p)$
defines $m \equiv a + q * d$
assumes $n > 0$ *coprime a d*
shows $[m = a]\ (mod\ d)$ **and** *coprime m n*
⟨*proof*⟩

Theorem 8.17

The case $d = 1$ is exactly the case described in *dcharacter ?n ?*$\chi \Longrightarrow$ *dcharacter.induced-modulus ?n ?*χ *1* = (*?*χ = *principal-dchar ?n*).

theorem (**in** *dcharacter*) *induced-modulus-altdef2*:
assumes *d dvd n* $d \neq 1$
defines $\chi_1 \equiv$ *principal-dchar n*
shows *induced-modulus d* $\longleftrightarrow (\exists\,\Phi.\ dcharacter\ d\ \Phi \wedge (\forall\,k.\ \chi\ k = \Phi\ k * \chi_1\ k))$
⟨*proof*⟩

7.4 The conductor of a character

context *dcharacter*
begin

definition *conductor* = *Min* {*d. induced-modulus d*}

lemma *conductor-fin*: *finite* {*d. induced-modulus d*}
⟨*proof*⟩

lemma *conductor-induced*: *induced-modulus conductor*
⟨*proof*⟩

lemma *conductor-le-iff*: *conductor* $\leq a \longleftrightarrow (\exists\,d \leq a.$ *induced-modulus d*)
⟨*proof*⟩

lemma *conductor-ge-iff*: *conductor* $\geq a \longleftrightarrow (\forall\,d.$ *induced-modulus d* $\longrightarrow d \geq a)$
⟨*proof*⟩

lemma *conductor-leI*: *induced-modulus d* $\Longrightarrow$ *conductor* $\leq d$
⟨*proof*⟩

lemma *conductor-geI*: $(\bigwedge d.$ *induced-modulus d* $\Longrightarrow d \geq a) \Longrightarrow$ *conductor* $\geq a$
⟨*proof*⟩

lemma *conductor-dvd*: *conductor dvd n*
⟨*proof*⟩

lemma *conductor-le-modulus*: *conductor* $\leq n$
⟨*proof*⟩

lemma *conductor-gr-0*: *conductor* > 0

⟨*proof*⟩

lemma *conductor-eq-1-iff-principal*: *conductor* = *1* ⟷ χ = *principal-dchar n*
⟨*proof*⟩

lemma *conductor-principal* [*simp*]: χ = *principal-dchar n* ⟹ *conductor* = *1*
⟨*proof*⟩

lemma *nonprimitive-imp-conductor-less*:
assumes ¬*primitive-dchar n* χ
shows *conductor* < *n*
⟨*proof*⟩

lemma (**in** *nonprimitive-dchar*) *conductor-less-modulus*: *conductor* < *n*
⟨*proof*⟩

Theorem 8.18

theorem *primitive-principal-form*:
defines χ_1 ≡ *principal-dchar n*
assumes χ ≠ *principal-dchar n*
shows ∃ Φ. *primitive-dchar conductor* Φ ∧ (∀ *n*. $\chi(n) = \Phi(n) * \chi_1(n)$)
⟨*proof*⟩

definition *primitive-extension* :: *nat* ⇒ *complex* **where**
primitive-extension =
(*SOME* Φ. *primitive-dchar conductor* Φ ∧ (∀ *k*. χ *k* = Φ *k* ∗ *principal-dchar n k*))

lemma
assumes *nonprincipal*: χ ≠ *principal-dchar n*
shows *primitive-primitive-extension*: *primitive-dchar conductor primitive-extension*
and *principal-decomposition*: χ *k* = *primitive-extension k* ∗ *principal-dchar n k*
⟨*proof*⟩

end

7.5 The connection between primitivity and separability

lemma *residue-mult-group-coset*:
fixes *m n m1 m2* :: *nat* **and** *f* :: *nat* ⇒ *nat* **and** *G H*
defines *G* ≡ *residue-mult-group n*
defines *H* ≡ *residue-mult-group m*
defines *f* ≡ (λ*k*. *k mod m*)
assumes *b* ∈ ($rcosets_G$ *kernel G H f*)
assumes *m1* ∈ *b m2* ∈ *b*
assumes *n* > *1 m dvd n*
shows *m1 mod m* = *m2 mod m*
⟨*proof*⟩

lemma *residue-mult-group-kernel-partition*:
fixes *m n* :: *nat* **and** *f* :: *nat* ⇒ *nat* **and** *G H*
defines *G* ≡ *residue-mult-group n*
defines *H* ≡ *residue-mult-group m*
defines *f* ≡ (λ*k*. *k mod m*)
assumes *m* > *1* *n* > *0* *m dvd n*
shows *partition* (*carrier G*) ($rcosets_G$ *kernel G H f*)
and *card* ($rcosets_G$ *kernel G H f*) = *totient m*
and *card* (*kernel G H f*) = *totient n div totient m*
and *b* ∈($rcosets_G$ *kernel G H f*) ⟹ *b* ≠ {}
and *b* ∈($rcosets_G$ *kernel G H f*) ⟹ *card* (*kernel G H f*) = *card b*
and *bij-betw* (λ*b*. (*the-elem* (*f* ' *b*))) ($rcosets_G$ *kernel G H f*) (*carrier H*)
⟨*proof*⟩

lemma *primitive-iff-separable-lemma*:
assumes *prod*: (∀ *n*. χ *n* = Φ *n* ∗ χ_1 *n*) ∧ *primitive-dchar d* Φ
assumes ‹*d* > *1*› ‹*0* < *k*› ‹*d dvd k*› ‹*k* > *1*›
shows (∑ *m* | *m* ∈ {*1..k*} ∧ *coprime m k*. Φ(*m*) ∗ *unity-root d m*) =
(*totient k div totient d*) ∗ (∑ *m* | *m* ∈ {*1..d*} ∧ *coprime m d*. Φ(*m*) ∗ *unity-root d m*)
⟨*proof*⟩

Theorem 8.19

theorem (**in** *dcharacter*) *primitive-iff-separable*:
primitive-dchar n χ ⟷ (∀ *k*>*0*. *separable k*)
⟨*proof*⟩

Theorem 8.20

theorem (**in** *primitive-dchar*) *fourier-primitive*:
includes *no vec-lambda-syntax*
fixes τ :: *complex*
defines τ ≡ *gauss-sum 1* / *sqrt n*
shows χ *m* = τ / *sqrt n* ∗ (∑ *k*=*1..n*. *cnj* (χ *k*) ∗ *unity-root n* (−*m*∗*k*))
and *norm* τ = *1*
⟨*proof*⟩

unbundle *vec-lambda-syntax*

end

8 The Pólya–Vinogradov Inequality

theory *Polya-Vinogradov*
imports
Gauss-Sums
Dirichlet-Series.Divisor-Count

begin

unbundle *no vec-lambda-syntax*

8.1 The case of primitive characters

We first prove a stronger variant of the Pólya–Vinogradov inequality for primitive characters. The fully general variant will then simply be a corollary of this. First, we need some bounds on logarithms, exponentials, and the harmonic numbers:

lemma *exp-1-less-powr*:
assumes $x > (0::real)$
shows $exp\ 1 < (1 + 1\ /\ x)\ powr\ (x+1)$
⟨*proof*⟩

lemma *harm-aux-ineq-1*:
fixes $k :: real$
assumes $k > 1$
shows $1\ /\ k < ln\ (1 + 1\ /\ (k - 1))$
⟨*proof*⟩

lemma *harm-aux-ineq-2-lemma*:
assumes $x \geq (0::real)$
shows $1 < (x + 1) * ln\ (1 + 2\ /\ (2 * x + 1))$
⟨*proof*⟩

lemma *harm-aux-ineq-2*:
fixes $k :: real$
assumes $k \geq 1$
shows $1\ /\ (k + 1) < ln\ (1 + 2\ /\ (2 * k + 1))$
⟨*proof*⟩

lemma *nat-0-1-induct* [*case-names 0 1 step*]:
assumes $P\ 0\ P\ 1\ \bigwedge n.\ n \geq 1 \Longrightarrow P\ n \Longrightarrow P\ (Suc\ n)$
shows $P\ n$
⟨*proof*⟩

lemma *harm-less-ln*:
fixes $m :: nat$
assumes $m > 0$
shows $harm\ m < ln\ (2 * m + 1)$
⟨*proof*⟩

Theorem 8.21

theorem (**in** *primitive-dchar*) *polya-vinogradov-inequality-primitive*:
fixes $x :: nat$
shows $norm\ (\sum m{=}1..x.\ \chi\ m) < sqrt\ n * ln\ n$
⟨*proof*⟩

8.2 General case

We now first prove the inequality for the general case in terms of the divisor function:

theorem (**in** *dcharacter*) *polya-vinogradov-inequality-explicit*:
  assumes *nonprincipal*: $\chi \neq$ *principal-dchar n*
  shows *norm* (*sum* χ {*1..x*}) < *sqrt conductor* ∗ *ln conductor* ∗ *divisor-count* (*n div conductor*)
⟨*proof*⟩

Next, we obtain a suitable upper bound on the number of divisors of n:

lemma *divisor-count-upper-bound-aux*:
  fixes *n* :: *nat*
  shows *divisor-count n* ≤ *2* ∗ *card* {*d. d dvd n* ∧ *d* ≤ *sqrt n*}
⟨*proof*⟩

lemma *divisor-count-upper-bound*:
  fixes *n* :: *nat*
  shows *divisor-count n* ≤ *2* ∗ *nat* ⌊*sqrt n*⌋
⟨*proof*⟩

lemma *divisor-count-upper-bound'*:
  fixes *n* :: *nat*
  shows *real* (*divisor-count n*) ≤ *2* ∗ *sqrt n*
⟨*proof*⟩

We are now ready to prove the 'regular' Pólya–Vinogradov inequality.

Apostol formulates it in the following way (Theorem 13.15, notation adapted): 'If χ is any nonprincipal character mod n, then for all $x \geq 2$ we have $\sum_{m \leq x} \chi(m) = O(\sqrt{n} \log n)$.'

The precondition $x \geq 2$ here is completely unnecessary. The 'Big-O' notation is somewhat problematic since it does not make explicit in what way the variables are quantified (in particular the x and the χ). The statement of the theorem in this way (for a fixed character χ) seems to suggest that n is fixed here, which would make the use of 'Big-O' completely vacuous, since it is an asymptotic statement about n.

We therefore decided to formulate the inequality in the following more explicit way, even giving an explicit constant factor:

theorem (**in** *dcharacter*) *polya-vinogradov-inequality*:
  assumes *nonprincipal*: $\chi \neq$ *principal-dchar n*
  shows *norm* ($\sum$ *m=1..x.* χ *m*) < *2* ∗ *sqrt n* ∗ *ln n*
⟨*proof*⟩

unbundle *vec-lambda-syntax*

end

References

[1] T. M. Apostol. *Introduction to Analytic Number Theory*. Undergraduate Texts in Mathematics. Springer-Verlag, 1976.