Dirichlet L-functions and Dirichlet's Theorem

Manuel Eberl

May 26, 2024

Abstract

This article provides a formalisation of Dirichlet characters and Dirichlet L-functions including proofs of their basic properties – most notably their analyticity, their areas of convergence, and their non-vanishing for $\mathfrak{R}(s) \geq 1$. All of this is built in a very high-level style using Dirichlet series. The proof of the non-vanishing follows a very short and elegant proof by Newman [4], which we attempt to reproduce faithfully in a similar level of abstraction in Isabelle.

This also leads to a relatively short proof of Dirichlet's Theorem, which states that, if h and n are coprime, there are infinitely many primes p with $p \equiv h \pmod{n}$.

Contents

1 Multiplicative Characters of Finite Abelian Groups

theory *Multiplicative-Characters*
imports
Complex-Main
Finitely-Generated-Abelian-Groups.Finitely-Generated-Abelian-Groups
begin

notation *integer-mod-group* (Z)

1.1 Definition of characters

A (multiplicative) character is a completely multiplicative function from a group to the complex numbers. For simplicity, we restrict this to finite abelian groups here, which is the most interesting case.

Characters form a group where the identity is the *principal* character that maps all elements to 1, multiplication is point-wise multiplication of the characters, and the inverse is the point-wise complex conjugate.

This group is often called the *Pontryagin dual* group and is isomorphic to the original group (in a non-natural way) while the double-dual group *is* naturally isomorphic to the original group.

To get extensionality of the characters, we also require characters to map anything that is not in the group to 0.

definition *principal-char* :: ($'a$, $'b$) *monoid-scheme* $\Rightarrow$ $'a$ $\Rightarrow$ *complex* **where**
principal-char G a = (*if a* $\in$ *carrier G then 1 else 0*)

definition *inv-character* **where**
inv-character χ = (λa. *cnj* (χ *a*))

lemma *inv-character-principal* [*simp*]: *inv-character* (*principal-char G*) = *principal-char G*
⟨*proof*⟩

lemma *inv-character-inv-character* [*simp*]: *inv-character* (*inv-character* χ) = χ
⟨*proof*⟩

lemma *eval-inv-character*: *inv-character* χ *j* = *cnj* (χ *j*)
⟨*proof*⟩

bundle *character-syntax*
begin
notation *principal-char* (χ_0ı)
end

locale *character* = *finite-comm-group* +
fixes χ :: $'a$ $\Rightarrow$ *complex*

assumes *char-one-nz*: χ **1** $\neq$ *0*
assumes *char-eq-0*: $a \notin$ *carrier G* $\Longrightarrow \chi$ *a* = *0*
assumes *char-mult* [*simp*]: $a \in$ *carrier G* $\Longrightarrow b \in$ *carrier G* $\Longrightarrow \chi$ ($a \otimes b$) = χ $a * \chi$ b
begin

1.2 Basic properties

lemma *char-one* [*simp*]: χ **1** = *1*
⟨*proof*⟩

lemma *char-power* [*simp*]: $a \in$ *carrier G* $\Longrightarrow \chi$ (*a* [^] *k*) = χ *a* ^ *k*
⟨*proof*⟩

lemma *char-root*:
assumes $a \in$ *carrier G*
shows χ *a* ^ *ord a* = *1*
⟨*proof*⟩

lemma *char-root'*:
assumes $a \in$ *carrier G*
shows χ *a* ^ *order G* = *1*
⟨*proof*⟩

lemma *norm-char*: *norm* (χ *a*) = (*if* $a \in$ *carrier G then 1 else 0*)
⟨*proof*⟩

lemma *char-eq-0-iff*: χ *a* = *0* $\longleftrightarrow a \notin$ *carrier G*
⟨*proof*⟩

lemma *inv-character*: *character G* (*inv-character* χ)
⟨*proof*⟩

lemma *mult-inv-character*: χ *k* $*$ *inv-character* χ *k* = *principal-char G k*
⟨*proof*⟩

lemma
assumes $a \in$ *carrier G*
shows *char-inv*: χ (*inv a*) = *cnj* (χ *a*) **and** *char-inv'*: χ (*inv a*) = *inverse* (χ *a*)
⟨*proof*⟩

end

lemma (**in** *finite-comm-group*) *character-principal* [*simp*, *intro*]: *character G* (*principal-char G*)
⟨*proof*⟩

lemmas [*simp,intro*] = *finite-comm-group.character-principal*

lemma *character-ext*:
  assumes *character* G χ *character* G χ' $\bigwedge x.\ x \in$ *carrier* $G \Longrightarrow \chi\ x = \chi'\ x$
  shows $\chi = \chi'$
⟨*proof*⟩

lemma *character-mult* [*intro*]:
  assumes *character* G χ *character* G χ'
  shows *character* G $(\lambda x.\ \chi\ x * \chi'\ x)$
⟨*proof*⟩

lemma *character-inv-character-iff* [*simp*]: *character* G (*inv-character* χ) $\longleftrightarrow$ *character* G χ
⟨*proof*⟩

definition *characters* :: $('a, 'b)$ *monoid-scheme* $\Rightarrow$ $('a \Rightarrow$ *complex*) *set* **where**
  characters $G = \{\chi.$ *character* G $\chi\}$

1.3 The Character group

The characters of a finite abelian group G form another group $\widehat{G}$, which is called its Pontryagin dual group. This generalises to the more general setting of locally compact abelian groups, but we restrict ourselves to the finite setting because it is much easier.

definition *Characters* :: $('a, 'b)$ *monoid-scheme* $\Rightarrow$ $('a \Rightarrow$ *complex*) *monoid*
  where *Characters* $G = (\!|$ *carrier* = *characters* G, *monoid.mult* = $(\lambda\chi_1\ \chi_2\ k.\ \chi_1\ k * \chi_2\ k)$,
    one = *principal-char* G $|\!)$

lemma *carrier-Characters*: *carrier* (*Characters* G) = *characters* G
  ⟨*proof*⟩

lemma *one-Characters*: *one* (*Characters* G) = *principal-char* G
  ⟨*proof*⟩

lemma *mult-Characters*: *monoid.mult* (*Characters* G) $\chi_1\ \chi_2 = (\lambda a.\ \chi_1\ a * \chi_2\ a)$
  ⟨*proof*⟩

context *finite-comm-group*
begin

sublocale *principal*: *character* G *principal-char* G ⟨*proof*⟩

lemma *finite-characters* [*intro*]: *finite* (*characters* G)
⟨*proof*⟩

lemma *finite-comm-group-Characters* [*intro*]: *finite-comm-group* (*Characters G*)
⟨*proof*⟩

end

lemma (**in** *character*) *character-in-order-1*:
  assumes *order G* = *1*
  shows χ = *principal-char G*
⟨*proof*⟩

lemma (**in** *finite-comm-group*) *characters-in-order-1*:
  assumes *order G* = *1*
  shows *characters G* = {*principal-char G*}
  ⟨*proof*⟩

lemma (**in** *character*) *inv-Characters*: $inv_{Characters\ G}\ \chi$ = *inv-character* χ
⟨*proof*⟩

lemma (**in** *finite-comm-group*) *inv-Characters′*:
  $\chi \in$ *characters G* $\Longrightarrow$ $inv_{Characters\ G}\ \chi$ = *inv-character* χ
  ⟨*proof*⟩

lemmas (**in** *finite-comm-group*) *Characters-simps* =
  carrier-Characters mult-Characters one-Characters inv-Characters′

lemma *inv-Characters′*: $\chi \in$ *characters G* $\Longrightarrow$ $inv_{Characters\ G}\ \chi$ = *inv-character* χ
  ⟨*proof*⟩

1.4 The isomorphism between a group and its dual

We start this section by inspecting the special case of a cyclic group. Here, any character is fixed by the value it assigns to the generating element of the cyclic group. This can then be used to construct a bijection between the nth unit roots and the elements of the character group - implying the other results.

lemma (**in** *finite-cyclic-group*)
  defines *ic*: *induce-char* ≡ (λ*c*::*complex*. (λ*a*. *if a*∈*carrier G then c powi get-exp gen a else 0*))
  shows *order-Characters*: *order* (*Characters G*) = *order G*
  and *gen-fixes-char*: ⟦*character G a*; *character G b*; *a gen* = *b gen*⟧ $\Longrightarrow$ *a* = *b*
  and *unity-root-induce-char*: *z* ^ *order G* = *1* $\Longrightarrow$ *character G* (*induce-char z*)
⟨*proof*⟩

Moreover, we can show that a character that assigns a "true" root of unity to the generating element of the group, generates the character group.

lemma (**in** *finite-cyclic-group*) *finite-cyclic-group-Characters*:
  obtains χ **where** *finite-cyclic-group* (*Characters G*) χ

⟨*proof*⟩

And as two cyclic groups of the same order are isomorphic it follows the isomorphism of a finite cyclic group and its dual.

lemma (**in** *finite-cyclic-group*) *Characters-iso*:
$G \cong$ *Characters G*
⟨*proof*⟩

The character groups of two isomorphic groups are also isomorphic.

lemma (**in** *finite-comm-group*) *iso-imp-iso-chars*:
assumes $G \cong H$ *group H*
shows *Characters G* $\cong$ *Characters H*
⟨*proof*⟩

The following two lemmas characterize the way a character behaves in a direct group product: a character on the product induces characters on each of the factors. Also, any character on the direct product can be decomposed into a pointwise product of characters on the factors.

lemma *DirProds-subchar*:
assumes *finite-comm-group* (*DirProds Gs I*)
and x: $x \in$ *carrier* (*Characters* (*DirProds Gs I*))
and i: $i \in I$
and I: *finite I*
defines g: $g \equiv (\lambda c.\ (\lambda i{\in}I.\ (\lambda a.\ c\ ((\lambda i{\in}I.\ \mathbf{1}_{Gs\ i})(i{:=}a)))))$
shows *character* (*Gs i*) (*g x i*)
⟨*proof*⟩

lemma *Characters-DirProds-single-prod*:
assumes *finite-comm-group* (*DirProds Gs I*)
and x: $x \in$ *carrier* (*Characters* (*DirProds Gs I*))
and I: *finite I*
defines g: $g \equiv (\lambda I.\ (\lambda c.\ (\lambda i{\in}I.\ (\lambda a.\ c\ ((\lambda i{\in}I.\ \mathbf{1}_{Gs\ i})(i{:=}a))))))$
shows (λe. *if* $e \in$*carrier*(*DirProds Gs I*) *then* $\prod i{\in}I.\ (g\ I\ x\ i)\ (e\ i)$ *else 0*) $= x$
(**is** *?g x* $= x$)
⟨*proof*⟩

This allows for the following: the character group of a direct product is isomorphic to the direct product of the character groups of the factors.

lemma (**in** *finite-comm-group*) *Characters-DirProds-iso*:
assumes *DirProds Gs I* $\cong G$ *group* (*DirProds Gs I*) *finite I*
shows *DirProds* (*Characters* $\circ$ *Gs*) $I \cong$ *Characters G*
⟨*proof*⟩

As thus both the group and its character group can be decomposed into the same cyclic factors, the isomorphism follows for any finite abelian group.

theorem (**in** *finite-comm-group*) *Characters-iso*:
shows $G \cong$ *Characters G*

⟨*proof*⟩

Hence, the orders are also equal.

corollary (**in** *finite-comm-group*) *order-Characters*:
order (*Characters G*) = *order G*
⟨*proof*⟩

corollary (**in** *finite-comm-group*) *card-characters*: *card* (*characters G*) = *order G*
⟨*proof*⟩

1.5 Non-trivial facts about characters

We characterize the character group of a quotient group as the group of characters that map all elements of the subgroup onto 1.

lemma (**in** *finite-comm-group*) *iso-Characters-FactGroup*:
assumes *H*: *subgroup H G*
shows (λχ *x*. *if x* ∈ *carrier G then* χ (*H* #> *x*) *else 0*) ∈
iso (*Characters* (*G Mod H*)) ((*Characters G*)⦇*carrier* := {χ∈*characters G*. ∀ *x*∈*H*. χ *x* = *1*}⦈)
⟨*proof*⟩

lemma (**in** *finite-comm-group*) *is-iso-Characters-FactGroup*:
assumes *H*: *subgroup H G*
shows *Characters* (*G Mod H*) ≅ (*Characters G*)⦇*carrier* := {χ∈*characters G*. ∀ *x*∈*H*. χ *x* = *1*}⦈
⟨*proof*⟩

In order to derive the number of extensions a character on a subgroup has to the entire group, we introduce the group homomorphism *restrict-char* that restricts a character to a given subgroup *H*.

definition *restrict-char*::′*a set* ⇒ (′*a* ⇒ *complex*) ⇒ (′*a* ⇒ *complex*) **where**
restrict-char H χ = (λ*e*. *if e*∈*H then* χ *e else 0*)

lemma (**in** *finite-comm-group*) *restrict-char-hom*:
assumes *subgroup H G*
shows *group-hom* (*Characters G*) (*Characters* (*G*⦇*carrier* := *H*⦈)) (*restrict-char H*)
⟨*proof*⟩

The kernel is just the set of the characters that are 1 on all of *H*.

lemma (**in** *finite-comm-group*) *restrict-char-kernel*:
assumes *subgroup H G*
shows *kernel* (*Characters G*) (*Characters* (*G*⦇*carrier* := *H*⦈)) (*restrict-char H*)
= {χ∈*characters G*. ∀ *x*∈*H*. χ *x* = *1*}
⟨*proof*⟩

Also, all of the characters on the subgroup are the image of some character on the whole group.

lemma (**in** *finite-comm-group*) *restrict-char-image*:
assumes *subgroup H G*
shows *restrict-char H ‘ (carrier (Characters G)) = carrier (Characters (G(|carrier := H|)))*
⟨*proof*⟩

It follows that any character on H can be extended to a character on G.

lemma (**in** *finite-comm-group*) *character-extension-exists*:
assumes *subgroup H G character (G(|carrier := H|)) χ*
obtains *χ′* **where** *character G χ′* **and** *⋀x. x ∈ H ⟹ χ′ x = χ x*
⟨*proof*⟩

For two characters on a group G the number of characters on subgroup H that share the values with them is the same for both.

lemma (**in** *finite-comm-group*) *character-restrict-card*:
assumes *subgroup H G character G a character G b*
shows *card {χ′∈characters G. ∀x∈H. χ′ x = a x} = card {χ′∈characters G. ∀x∈H. χ′ x = b x}*
⟨*proof*⟩

These lemmas allow to show that the number of extensions of a character on H to a character on G is just $|G|/|H|$.

theorem (**in** *finite-comm-group*) *card-character-extensions*:
assumes *subgroup H G character (G(|carrier := H|)) χ*
shows *card {χ′∈characters G. ∀x∈H. χ′ x = χ x} * card H = order G*
⟨*proof*⟩

Lastly, we can also show that for each $x \in H$ of order $n > 1$ and each n-th root of unity z, there exists a character χ on G such that $\chi(x) = z$.

lemma (**in** *group*) *powi-get-exp-self*:
fixes *z::complex*
assumes *z ^ n = 1 x ∈ carrier G ord x = n n > 1*
shows *z powi get-exp x x = z*
⟨*proof*⟩

corollary (**in** *finite-comm-group*) *character-with-value-exists*:
assumes *x ∈ carrier G* **and** *x ≠* **1** **and** *z ^ ord x = 1*
obtains *χ* **where** *character G χ* **and** *χ x = z*
⟨*proof*⟩

In particular, for any x that is not the identity element, there exists a character χ such that $\chi(x) \neq 1$.

corollary (**in** *finite-comm-group*) *character-neq-1-exists*:
assumes *x ∈ carrier G* **and** *x ≠* **1**
obtains *χ* **where** *character G χ* **and** *χ x ≠ 1*
⟨*proof*⟩

1.6 The first orthogonality relation

The entries of any non-principal character sum to 0.

theorem (**in** *character*) *sum-character*:
$(\sum x{\in}carrier\ G.\ \chi\ x) = (if\ \chi = principal\text{-}char\ G\ then\ of\text{-}nat\ (order\ G)\ else\ 0)$
⟨*proof*⟩

corollary (**in** *finite-comm-group*) *character-orthogonality1*:
assumes *character G* χ **and** *character G* χ'
shows $(\sum x{\in}carrier\ G.\ \chi\ x * cnj\ (\chi'\ x)) = (if\ \chi = \chi'\ then\ of\text{-}nat\ (order\ G)\ else\ 0)$
⟨*proof*⟩

1.7 The isomorphism between a group and its double dual

Lastly, we show that the double dual of a finite abelian group is naturally isomorphic to the original group via the obvious isomorphism $x \mapsto (\chi \mapsto \chi(x))$. It is easy to see that this is a homomorphism and that it is injective. The fact $|\widehat{\widehat{G}}| = |\widehat{G}| = |G|$ then shows that it is also surjective.

context *finite-comm-group*
begin

definition *double-dual-iso* :: $'a \Rightarrow ('a \Rightarrow complex) \Rightarrow complex$ **where**
double-dual-iso $x = (\lambda\chi.\ if\ character\ G\ \chi\ then\ \chi\ x\ else\ 0)$

lemma *double-dual-iso-apply* [*simp*]: *character G* $\chi \Longrightarrow$ *double-dual-iso* $x\ \chi = \chi\ x$
⟨*proof*⟩

lemma *character-double-dual-iso* [*intro*]:
assumes *x*: $x \in carrier\ G$
shows *character* (*Characters G*) (*double-dual-iso x*)
⟨*proof*⟩

lemma *double-dual-iso-mult* [*simp*]:
assumes $x \in carrier\ G\ y \in carrier\ G$
shows *double-dual-iso* $(x \otimes y) =$
double-dual-iso $x \otimes_{Characters\ (Characters\ G)}$ *double-dual-iso y*
⟨*proof*⟩

lemma *double-dual-iso-one* [*simp*]:
double-dual-iso **1** = *principal-char* (*Characters G*)
⟨*proof*⟩

lemma *inj-double-dual-iso*: *inj-on double-dual-iso* (*carrier G*)
⟨*proof*⟩

lemma *double-dual-iso-eq-iff* [*simp*]:

$x \in$ *carrier* $G \Longrightarrow y \in$ *carrier* $G \Longrightarrow$ *double-dual-iso* $x =$ *double-dual-iso* $y \longleftrightarrow x = y$
⟨*proof*⟩

theorem *double-dual-iso*: *double-dual-iso* $\in$ *iso* G (*Characters* (*Characters* G))
⟨*proof*⟩

lemma *double-dual-is-iso*: *Characters* (*Characters* G) $\cong G$
⟨*proof*⟩

The second orthogonality relation follows from the first one via Pontryagin duality:

theorem *sum-characters*:
assumes x: $x \in$ *carrier* G
shows $(\sum \chi \in$ *characters* $G.\ \chi\ x) = ($*if* $x = \mathbf{1}$ *then of-nat* (*order* G) *else 0*)
⟨*proof*⟩

corollary *character-orthogonality2*:
assumes $x \in$ *carrier* G $y \in$ *carrier* G
shows $(\sum \chi \in$ *characters* $G.\ \chi\ x *$ *cnj* $(\chi\ y)) = ($*if* $x = y$ *then of-nat* (*order* G) *else 0*)
⟨*proof*⟩

end

no-notation *integer-mod-group* (Z)
end

2 Dirichlet Characters

theory *Dirichlet-Characters*
imports
Multiplicative-Characters
HOL−Number-Theory.Residues
Dirichlet-Series.Multiplicative-Function
begin

Dirichlet characters are essentially just the characters of the multiplicative group of integer residues $\mathbb{ZZ}/n\mathbb{ZZ}$ for some fixed n. For convenience, these residues are usually represented by natural numbers from 0 to $n-1$, and we extend the characters to all natural numbers periodically, so that $\chi(k \bmod n) = \chi(k)$ holds.

Numbers that are not coprime to n are not in the group and therefore are assigned 0 by all characters.

2.1 The multiplicative group of residues

definition *residue-mult-group* :: *nat* $\Rightarrow$ *nat monoid* **where**

residue-mult-group n = ⦇ *carrier* = *totatives n*, *monoid.mult* = ($\lambda x\ y.$ ($x * y$) *mod n*), *one* = *1* ⦈

definition *principal-dchar* :: *nat* ⇒ *nat* ⇒ *complex* **where**
principal-dchar n = ($\lambda k.$ *if coprime k n then 1 else 0*)

lemma *principal-dchar-coprime* [*simp*]: *coprime k n* ⟹ *principal-dchar n k* = *1*
and *principal-dchar-not-coprime* [*simp*]: ¬*coprime k n* ⟹ *principal-dchar n k* = *0*
⟨*proof*⟩

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

lemma *principal-dchar-minus1* [*simp*]:
assumes *n* > *0*
shows *principal-dchar n* (*n* − *Suc 0*) = *1*
⟨*proof*⟩

lemma *mod-in-totatives*: *n* > *1* ⟹ *a mod n* ∈ *totatives n* ⟷ *coprime a n*
⟨*proof*⟩

bundle *dcharacter-syntax*
begin
notation *principal-dchar* (χ_0ı)
end

locale *residues-nat* =
fixes *n* :: *nat* (**structure**) **and** *G*
assumes *n*: *n* > *1*
defines *G* ≡ *residue-mult-group n*
begin

lemma *order* [*simp*]: *order G* = *totient n*
⟨*proof*⟩

lemma *totatives-mod* [*simp*]: *x* ∈ *totatives n* ⟹ *x mod n* = *x*
⟨*proof*⟩

lemma *principal-dchar-minus1* [*simp*]: *principal-dchar n* (*n* − *Suc 0*) = *1*
⟨*proof*⟩

sublocale *finite-comm-group G*
⟨*proof*⟩

2.2 Definition of Dirichlet characters

The following two functions make the connection between Dirichlet characters and the multiplicative characters of the residue group.

definition *c2dc* :: (*nat* ⇒ *complex*) ⇒ (*nat* ⇒ *complex*) **where**
c2dc χ = (λ*x*. χ (*x mod n*))

definition *dc2c* :: (*nat* ⇒ *complex*) ⇒ (*nat* ⇒ *complex*) **where**
dc2c χ = (λ*x*. *if x* < *n then* χ *x else 0*)

lemma *dc2c-c2dc* [*simp*]:
assumes *character G* χ
shows *dc2c* (*c2dc* χ) = χ
⟨*proof*⟩

end

locale *dcharacter* = *residues-nat* +
fixes χ :: *nat* ⇒ *complex*
assumes *mult-aux*: *a* ∈ *totatives n* ⟹ *b* ∈ *totatives n* ⟹ χ (*a* ∗ *b*) = χ *a* ∗ χ *b*
assumes *eq-zero*: ¬*coprime a n* ⟹ χ *a* = *0*
assumes *periodic*: χ (*a* + *n*) = χ *a*
assumes *one-not-zero*: χ *1* ≠ *0*
begin

lemma *zero-eq-0* [*simp*]: χ *0* = *0*
⟨*proof*⟩

lemma *Suc-0* [*simp*]: χ (*Suc 0*) = *1*
⟨*proof*⟩

lemma *periodic-mult*: χ (*a* + *m* ∗ *n*) = χ *a*
⟨*proof*⟩

lemma *minus-one-periodic* [*simp*]:
assumes *k* > *0*
shows χ (*k* ∗ *n* − *1*) = χ (*n* − *1*)
⟨*proof*⟩

lemma *cong*:
assumes [*a* = *b*] (*mod n*)
shows χ *a* = χ *b*
⟨*proof*⟩

lemma *mod* [*simp*]: χ (*a mod n*) = χ *a*
⟨*proof*⟩

lemma *mult* [*simp*]: χ (*a* ∗ *b*) = χ *a* ∗ χ *b*
⟨*proof*⟩

sublocale *mult*: *completely-multiplicative-function* χ
⟨*proof*⟩

lemma *eq-zero-iff*: $\chi\ x = 0 \longleftrightarrow \neg coprime\ x\ n$
⟨*proof*⟩

lemma *minus-one'*: $\chi\ (n - 1) \in \{-1, 1\}$
⟨*proof*⟩

lemma *c2dc-dc2c* [*simp*]: *c2dc* (*dc2c* χ) = χ
⟨*proof*⟩

lemma *character-dc2c*: *character G* (*dc2c* χ)
⟨*proof*⟩

sublocale *dc2c*: *character G dc2c* χ
⟨*proof*⟩

lemma *dcharacter-inv-character* [*intro*]: *dcharacter n* (*inv-character* χ)
⟨*proof*⟩

lemma *norm*: *norm* (χ *k*) = (*if coprime k n then 1 else 0*)
⟨*proof*⟩

lemma *norm-le-1*: *norm* (χ *k*) $\leq$ *1*
⟨*proof*⟩

end

definition *dcharacters* :: *nat* ⇒ (*nat* ⇒ *complex*) *set* **where**
dcharacters n = {χ. *dcharacter n* χ}

context *residues-nat*
begin

lemma *character-dc2c*: *dcharacter n* $\chi \Longrightarrow$ *character G* (*dc2c* χ)
⟨*proof*⟩

lemma *dcharacter-c2dc*:
assumes *character G* χ
shows *dcharacter n* (*c2dc* χ)
⟨*proof*⟩

lemma *principal-dchar-altdef*: *principal-dchar n* = *c2dc* (*principal-char G*)
⟨*proof*⟩

sublocale *principal*: *dcharacter n G principal-dchar n*
⟨*proof*⟩

lemma *c2dc-principal* [*simp*]: *c2dc* (*principal-char G*) = *principal-dchar n*

⟨*proof*⟩

lemma *dc2c-principal* [*simp*]: *dc2c* (*principal-dchar n*) = *principal-char G*
⟨*proof*⟩

lemma *bij-betw-dcharacters-characters*:
bij-betw dc2c (*dcharacters n*) (*characters G*)
⟨*proof*⟩

lemma *bij-betw-characters-dcharacters*:
bij-betw c2dc (*characters G*) (*dcharacters n*)
⟨*proof*⟩

lemma *finite-dcharacters* [*intro*]: *finite* (*dcharacters n*)
⟨*proof*⟩

lemma *card-dcharacters* [*simp*]: *card* (*dcharacters n*) = *totient n*
⟨*proof*⟩

end

lemma *inv-character-eq-principal-dchar-iff* [*simp*]:
inv-character χ = *principal-dchar n* $\longleftrightarrow$ χ = *principal-dchar n*
⟨*proof*⟩

2.3 Sums of Dirichlet characters

lemma (**in** *dcharacter*) *sum-dcharacter-totatives*:
($\sum$*x*∈*totatives n*. χ *x*) = (*if* χ = *principal-dchar n then of-nat* (*totient n*) *else* *0*)
⟨*proof*⟩

lemma (**in** *dcharacter*) *sum-dcharacter-block*:
($\sum$*x*<*n*. χ *x*) = (*if* χ = *principal-dchar n then of-nat* (*totient n*) *else 0*)
⟨*proof*⟩

lemma (**in** *dcharacter*) *sum-dcharacter-block'*:
sum χ {*Suc 0..n*} = (*if* χ = *principal-dchar n then of-nat* (*totient n*) *else 0*)
⟨*proof*⟩

lemma (**in** *dcharacter*) *sum-lessThan-dcharacter*:
assumes $\chi \neq$ *principal-dchar n*
shows ($\sum$*x*<*m*. χ *x*) = ($\sum$*x*<*m mod n*. χ *x*)
⟨*proof*⟩

lemma (**in** *dcharacter*) *sum-dcharacter-lessThan-le*:
assumes $\chi \neq$ *principal-dchar n*
shows *norm* ($\sum$*x*<*m*. χ *x*) $\leq$ *totient n*

⟨*proof*⟩

lemma (**in** *dcharacter*) *sum-dcharacter-atMost-le*:
  assumes $\chi \neq$ *principal-dchar n*
  shows *norm* ($\sum x \leq m.\ \chi\ x$) $\leq$ *totient n*
  ⟨*proof*⟩

lemma (**in** *residues-nat*) *sum-dcharacters*:
  ($\sum \chi \in$*dcharacters n*. χ *x*) = (*if* [*x* = *1*] (*mod n*) *then of-nat* (*totient n*) *else 0*)
⟨*proof*⟩

lemma (**in** *dcharacter*) *even-dcharacter-linear-sum-eq-0* [*simp*]:
  assumes $\chi \neq$ *principal-dchar n* **and** χ (*n* − *1*) = *1*
  shows ($\sum$ *k*=*Suc 0*..<*n*. *of-nat k* ∗ χ *k*) = *0*
⟨*proof*⟩

end

3 Dirichlet L-functions

theory *Dirichlet-L-Functions*
imports
  Dirichlet-Characters
  HOL−Library.Landau-Symbols
  Zeta-Function.Zeta-Function
begin

We can now define the Dirichlet L-functions. These are essentially the functions in the complex plane that the Dirichlet series $\sum_{k=1}^{\infty} \chi(k)k^{-s}$ converge to, for some fixed Dirichlet character χ.

First of all, we need to take care of a syntactical problem: The notation for vectors uses χ as syntax, which causes some annoyance to us, so we disable it locally.

3.1 Definition and basic properties

We now define Dirichlet L functions as a finite linear combination of Hurwitz ζ functions. This has the advantage that we directly get the analytic continuation over the full domain and only need to prove that the series really converges to this definition whenever it does converge, which is not hard to do.

definition *Dirichlet-L* :: *nat* ⇒ (*nat* ⇒ *complex*) ⇒ *complex* ⇒ *complex* **where**
  Dirichlet-L m χ *s* =
    (*if s* = *1 then*
      if χ = *principal-dchar m then 0 else eval-fds* (*fds* χ) *1*
    else
      of-nat m powr − *s* ∗ ($\sum$ *k* = *1*..*m*. χ *k* ∗ *hurwitz-zeta* (*real k* / *real m*) *s*))

lemma *Dirichlet-L-conv-hurwitz-zeta-nonprincipal*:
assumes $s \neq 1$
shows *Dirichlet-L n* χ $s =$
of-nat n powr $-s *$ ($\sum$ $k = 1..n.$ χ $k *$ *hurwitz-zeta* (*real k* / *real n*) *s*)
⟨*proof*⟩

Analyticity everywhere except 1 is trivial by the above definition, since the Hurwitz ζ function is analytic everywhere except 1. For L functions of non principal characters, we will have to show the analyticity at 1 separately later.

lemma *holomorphic-Dirichlet-L-weak*:
assumes $m > 0$ $1 \notin A$
shows *Dirichlet-L m* χ *holomorphic-on A*
⟨*proof*⟩

context *dcharacter*
begin

For a real value greater than 1, the formal Dirichlet series of an L function for some character χ converges to the L function.

lemma
fixes *s* :: *complex*
assumes *s*: *Re s* > 1
shows *abs-summable-Dirichlet-L*: *summable* ($\lambda n.$ *norm* (χ $n *$ *of-nat n powr* $-s$))
and *summable-Dirichlet-L*: *summable* ($\lambda n.$ χ $n *$ *of-nat n powr* $-s$)
and *sums-Dirichlet-L*: ($\lambda n.$ χ $n * n$ *powr* $-s$) *sums Dirichlet-L n* χ *s*
and *Dirichlet-L-conv-eval-fds-weak*: *Dirichlet-L n* χ $s =$ *eval-fds* (*fds* χ) *s*
⟨*proof*⟩

lemma *fds-abs-converges-weak*: *Re s* $> 1 \Longrightarrow$ *fds-abs-converges* (*fds* χ) *s*
⟨*proof*⟩

lemma *abs-conv-abscissa-weak*: *abs-conv-abscissa* (*fds* χ) ≤ 1
⟨*proof*⟩

Dirichlet L functions have the Euler product expansion

$$L(\chi, s) = \prod_p \left(1 - \frac{\chi(p)}{p^{-s}}\right)$$

for all s with $\Re(s) > 1$.

lemma
fixes *s* :: *complex* **assumes** *s*: *Re s* > 1
shows *Dirichlet-L-euler-product-LIMSEQ*:
($\lambda n.$ $\prod p \leq n.$ *if prime p then inverse* ($1 - \chi$ *p* / *nat-power p s*) *else 1*)
$\longrightarrow$ *Dirichlet-L n* χ *s* (**is** *?th1*)

and *Dirichlet-L-abs-convergent-euler-product*:
abs-convergent-prod (λp. *if prime p then inverse* ($1 - \chi\ p\ /\ p$ *powr s*) *else 1*)
(**is** *?th2*)
⟨*proof*⟩

lemma *Dirichlet-L-Re-gt-1-nonzero*:
assumes *Re s* > 1
shows *Dirichlet-L n* χ *s* $\neq 0$
⟨*proof*⟩

lemma *sum-dcharacter-antimono-bound*:
fixes *x0 a b* :: *real* **and** *f f′* :: *real* ⇒ *real*
assumes *nonprincipal*: $\chi \neq \chi_0$
assumes *x0*: *x0* ≥ 0 **and** *ab*: *x0* $\leq$ *a a* $<$ *b*
assumes *f′*: $\bigwedge x$. $x \geq x0 \Longrightarrow$ (*f has-field-derivative f′ x*) (*at x*)
assumes *f-nonneg*: $\bigwedge x$. $x \geq x0 \Longrightarrow f\ x \geq 0$
assumes *f′-nonpos*: $\bigwedge x$. $x \geq x0 \Longrightarrow f'\ x \leq 0$
shows *norm* ($\sum n \in$*real* −‘ {*a*<..*b*}. χ *n* ∗ (*f* (*real n*))) $\leq$ *2* ∗ *real* (*totient n*) ∗ *f a*
⟨*proof*⟩

lemma *summable-dcharacter-antimono*:
fixes *x0 a b* :: *real* **and** *f f′* :: *real* ⇒ *real*
assumes *nonprincipal*: $\chi \neq \chi_0$
assumes *f′*: $\bigwedge x$. $x \geq x0 \Longrightarrow$ (*f has-field-derivative f′ x*) (*at x*)
assumes *f-nonneg*: $\bigwedge x$. $x \geq x0 \Longrightarrow f\ x \geq 0$
assumes *f′-nonpos*: $\bigwedge x$. $x \geq x0 \Longrightarrow f'\ x \leq 0$
assumes *lim*: (*f* ⟶ *0*) *at-top*
shows *summable* (λn. χ *n* ∗ *f n*)
⟨*proof*⟩

lemma *conv-abscissa-le-0*:
fixes *s* :: *real*
assumes *nonprincipal*: $\chi \neq \chi_0$
shows *conv-abscissa* (*fds* χ) ≤ 0
⟨*proof*⟩

lemma *summable-Dirichlet-L′*:
assumes *nonprincipal*: $\chi \neq \chi_0$
assumes *s*: *Re s* > 0
shows *summable* (λn. χ *n* ∗ *of-nat n powr* −*s*)
⟨*proof*⟩

lemma
assumes $\chi \neq \chi_0$
shows *Dirichlet-L-conv-eval-fds*: $\bigwedge s$. *Re s* $> 0 \Longrightarrow$ *Dirichlet-L n* χ *s* = *eval-fds* (*fds* χ) *s*
and *holomorphic-Dirichlet-L*: *Dirichlet-L n* χ *holomorphic-on A*

⟨*proof*⟩

lemma *cnj-Dirichlet-L*:
cnj (*Dirichlet-L n* χ *s*) = *Dirichlet-L n* (*inv-character* χ) (*cnj s*)
⟨*proof*⟩**end**

lemma *holomorphic-Dirichlet-L* [*holomorphic-intros*]:
assumes *n* > *1* $\chi \neq$ *principal-dchar n* ∧ *dcharacter n* χ ∨ χ = *principal-dchar n* ∧ *1* ∉ *A*
shows *Dirichlet-L n* χ *holomorphic-on A*
⟨*proof*⟩

lemma *holomorphic-Dirichlet-L′* [*holomorphic-intros*]:
assumes *n* > *1 f holomorphic-on A*
$\chi \neq$ *principal-dchar n* ∧ *dcharacter n* χ ∨ χ = *principal-dchar n* ∧ (∀ *x*∈*A*. *f x* ≠ *1*)
shows (λ*s*. *Dirichlet-L n* χ (*f s*)) *holomorphic-on A*
⟨*proof*⟩

lemma *continuous-on-Dirichlet-L*:
assumes *n* > *1* $\chi \neq$ *principal-dchar n* ∧ *dcharacter n* χ ∨ χ = *principal-dchar n* ∧ *1* ∉ *A*
shows *continuous-on A* (*Dirichlet-L n* χ)
⟨*proof*⟩

lemma *continuous-on-Dirichlet-L′* [*continuous-intros*]:
assumes *continuous-on A f n* > *1*
and $\chi \neq$ *principal-dchar n* ∧ *dcharacter n* χ ∨ χ = *principal-dchar n* ∧ (∀ *x*∈*A*. *f x* ≠ *1*)
shows *continuous-on A* (λ*x*. *Dirichlet-L n* χ (*f x*))
⟨*proof*⟩

corollary *continuous-Dirichlet-L* [*continuous-intros*]:
n > *1* ⟹ $\chi \neq$ *principal-dchar n* ∧ *dcharacter n* χ ∨ χ = *principal-dchar n* ∧ *s* ≠ *1* ⟹
continuous (*at s within A*) (*Dirichlet-L n* χ)
⟨*proof*⟩

corollary *continuous-Dirichlet-L′* [*continuous-intros*]:
n > *1* ⟹ *continuous* (*at s within A*) *f* ⟹
$\chi \neq$ *principal-dchar n* ∧ *dcharacter n* χ ∨ χ = *principal-dchar n* ∧ *f s* ≠ *1* ⟹
continuous (*at s within A*) (λ*x*. *Dirichlet-L n* χ (*f x*))
⟨*proof*⟩

context *residues-nat*
begin

Applying the above to the $L(\chi_0, s)$, the L function of the principal character, we find that it differs from the Riemann ζ function only by multiplication with a constant that depends only on the modulus n. They therefore have the same analytic properties as the ζ function itself.

lemma *Dirichlet-L-principal*:
  fixes *s* :: *complex*
  shows *Dirichlet-L n* χ_0 *s* = ($\prod$ *p* | *prime p* ∧ *p dvd n*. (*1* − *1* / *p powr s*)) ∗ *zeta s*
    (**is** *?f s* = *?g s*)
⟨*proof*⟩**end**

3.2 The non-vanishing for $\Re(s) \geq 1$

lemma *coprime-prime-exists*:
  assumes *n* > (*0* :: *nat*)
  obtains *p* **where** *prime p coprime p n*
⟨*proof*⟩

The case of the principal character is trivial, since it differs from the Riemann $\zeta(s)$ only in a multiplicative factor that is clearly non-zero for $\Re(s) \geq 1$.

theorem (**in** *residues-nat*) *Dirichlet-L-Re-ge-1-nonzero-principal*:
  assumes *Re s* ≥ *1 s* ≠ *1*
  shows *Dirichlet-L n* (*principal-dchar n*) *s* ≠ *0*
⟨*proof*⟩

The proof for non-principal character is quite involved and is typically very complicated and technical in most textbooks. For instance, Apostol [1] proves the result separately for real and non-real characters, where the non-real case is relatively short and nice, but the real case involves a number of complicated asymptotic estimates.

The following proof, on the other hand – like our proof of the analogous result for the Riemann ζ function – is based on Newman's book [4]. Newman gives a very short, concise, and high-level sketch that we aim to reproduce faithfully here.

context *dcharacter*
begin
theorem *Dirichlet-L-Re-ge-1-nonzero-nonprincipal*:
  assumes $\chi \neq \chi_0$ **and** *Re u* ≥ *1*
  shows *Dirichlet-L n* χ *u* ≠ *0*
⟨*proof*⟩
  include *dcharacter-syntax*
  ⟨*proof*⟩

3.3 Asymptotic bounds on partial sums of Dirichlet L functions

The following are some bounds on partial sums of the L-function of a character that are useful for asymptotic reasoning, particularly for Dirichlet's

Theorem.

lemma *sum-upto-dcharacter-le*:
  assumes $\chi \neq \chi_0$
  shows *norm* (*sum-upto* χ *x*) $\leq$ *totient n*
⟨*proof*⟩

lemma *Dirichlet-L-minus-partial-sum-bound*:
  fixes *s* :: *complex* **and** *x* :: *real*
  assumes $\chi \neq \chi_0$ **and** *Re s* > *0* **and** *x* > *0*
  defines $\sigma \equiv$ *Re s*
  shows *norm* (*sum-upto* ($\lambda n.$ χ *n* ∗ *n powr* −*s*) *x* − *Dirichlet-L n* χ *s*) $\leq$
    real (*totient n*) ∗ (*2* + *cmod s* / σ) / *x powr* σ
⟨*proof*⟩

lemma *partial-Dirichlet-L-sum-bigo*:
  fixes *s* :: *complex* **and** *x* :: *real*
  assumes $\chi \neq \chi_0$ *Re s* > *0*
  shows ($\lambda x.$ *sum-upto* ($\lambda n.$ χ *n* ∗ *n powr* −*s*) *x* − *Dirichlet-L n* χ *s*) ∈ *O*($\lambda x.$ *x powr* −*s*)
⟨*proof*⟩**end**

3.4 Evaluation of $L(\chi, 0)$

context *residues-nat*
begin
lemma *Dirichlet-L-0-principal* [*simp*]: *Dirichlet-L n* χ_0 *0* = *0*
⟨*proof*⟩

end
context *dcharacter*
begin
lemma *Dirichlet-L-0-nonprincipal*:
  assumes *nonprincipal*: $\chi \neq \chi_0$
  shows *Dirichlet-L n* χ *0* = −($\sum$ *k*=*1*..<*n*. *of-nat k* ∗ χ *k*) / *of-nat n*
⟨*proof*⟩

lemma *Dirichlet-L-0-even* [*simp*]:
  assumes χ (*n* − *1*) = *1*
  shows *Dirichlet-L n* χ *0* = *0*
⟨*proof*⟩

lemma *Dirichlet-L-0*:
  Dirichlet-L n χ *0* = (*if* χ (*n* − *1*) = *1 then 0 else* −($\sum$ *k*=*1*..<*n*. *of-nat k* ∗ χ *k*) / *of-nat n*)
  ⟨*proof*⟩**end**

3.5 Properties of $L(\chi, s)$ for real χ

locale *real-dcharacter* = *dcharacter* +
  assumes *real*: χ *k* $\in \mathbb{R}$
begin

lemma *Im-eq-0* [*simp*]: *Im* (χ *k*) = *0*
  ⟨*proof*⟩

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

lemma *char-cases*: χ *k* $\in$ {−*1*, *0*, *1*}
⟨*proof*⟩

lemma *cnj* [*simp*]: *cnj* (χ *k*) = χ *k*
  ⟨*proof*⟩

lemma *inv-character-id* [*simp*]: *inv-character* χ = χ
  ⟨*proof*⟩

lemma *Dirichlet-L-in-Reals*:
  assumes *s* $\in \mathbb{R}$
  shows *Dirichlet-L n* χ *s* $\in \mathbb{R}$
⟨*proof*⟩

The following property of real characters is used by Apostol to show the non-vanishing of $L(\chi, 1)$. We have already shown this in a much easier way, but this particular result is still of general interest.

lemma
  assumes *k*: *k* > *0*
  shows *sum-char-divisors-ge*: *Re* ($\sum$ *d* | *d dvd k*. χ *d*) $\geq$ *0* (**is** *Re* (*?A k*) $\geq$ *0*)
  and *sum-char-divisors-square-ge*: *is-square k* $\Longrightarrow$ *Re* ($\sum$ *d* | *d dvd k*. χ *d*) $\geq$ *1*
⟨*proof*⟩

end

end

4 Dirichlet's Theorem on primes in arithmetic progressions

theory *Dirichlet-Theorem*
imports
  Dirichlet-L-Functions
  Bertrands-Postulate.Bertrand
  Landau-Symbols.Landau-More
begin

We can now turn to the proof of the main result: Dirichlet's theorem about the infinitude of primes in arithmetic progressions.

There are previous proofs of this by John Harrison in HOL Light [3] and by Mario Carneiro in Metamath [2]. Both of them strive to prove Dirichlet's theorem with a minimum amount of auxiliary results and definitions, whereas our goal was to get a short and simple proof of Dirichlet's theorem built upon a large library of Analytic Number Theory.

At this point, we already have the key part – the non-vanishing of $L(1, \chi)$ – and the proof was relatively simple and straightforward due to the large amount of Complex Analysis and Analytic Number Theory we have available. The remainder will be a bit more concrete, but still reasonably concise.

First, we need to re-frame some of the results from the AFP entry about Bertrand's postulate a little bit.

4.1 Auxiliary results

The AFP entry for Bertrand's postulate already provides a slightly stronger version of this for integer values of x, but we can easily extend this to real numbers to obtain a slightly nicer presentation.

lemma *sum-upto-mangoldt-le*:
  assumes $x \geq 0$
  shows *sum-upto mangoldt* $x \leq 3 / 2 * x$
⟨*proof*⟩

We can also, similarly, use the results from the Bertrand's postulate entry to show that the sum of $\ln p/p$ over all primes grows logarithmically.

lemma *Mertens-bigo*:
  $(\lambda x.\ (\sum p \mid \mathit{prime}\ p \wedge \mathit{real}\ p \leq x.\ \mathit{ln}\ p\ /\ p) - \mathit{ln}\ x) \in O(\lambda\text{-}.\ 1)$
⟨*proof*⟩

4.2 The contribution of the non-principal characters

The estimates in the next two sections are partially inspired by John Harrison's proof of Dirichlet's Theorem [3].

We first estimate the growth of the partial sums of

$$-L'(1, \chi)/L(1, \chi) = \sum_{k=1}^{\infty} \chi(k) \frac{\Lambda(k)}{k}$$

for a non-principal character χ and show that they are, in fact, bounded, which is ultimately a consequence of the non-vanishing of $L(1, \chi)$ for non-principal χ.

context *dcharacter*
begin

context
includes *no-vec-lambda-notation dcharacter-syntax*
fixes L
assumes *nonprincipal*: $\chi \neq \chi_0$
defines $L \equiv$ *Dirichlet-L n* χ *1*
begin

lemma *Dirichlet-L-nonprincipal-mangoldt-bound-aux-strong*:
assumes *x*: $x > 0$
shows *norm* (L ∗ *sum-upto* (λk. χ k ∗ *mangoldt* k / k) x − *sum-upto* (λk. χ k ∗ *ln* k / k) x)
≤ *9* / *2* ∗ *real* (*totient* n)
⟨*proof*⟩

lemma *Dirichlet-L-nonprincipal-mangoldt-aux-bound*:
(λx. L ∗ *sum-upto* (λk. χ k ∗ *mangoldt* k / k) x − *sum-upto* (λk. χ k ∗ *ln* k / k) x) ∈ $O(\lambda$-. *1*)
⟨*proof*⟩

lemma *nonprincipal-mangoldt-bound*:
(λx. *sum-upto* (λk. χ k ∗ *mangoldt* k / k) x) ∈ $O(\lambda$-. *1*) (**is** *?lhs* ∈ -)
⟨*proof*⟩

end
end

4.3 The contribution of the principal character

Next, we turn to the analogous partial sum for the principal character and show that it grows logarithmically and therefore is the dominant contribution.

context *residues-nat*
begin
context
includes *no-vec-lambda-notation dcharacter-syntax*
begin

lemma *principal-dchar-sum-bound*:
(λx. ($\sum p$ | *prime* p ∧ *real* $p \leq x$. χ_0 p ∗ (*ln* p / p)) − *ln* x) ∈ $O(\lambda$-. *1*)
⟨*proof*⟩

lemma *principal-dchar-sum-bound'*:
(λx. *sum-upto* (λk. χ_0 k ∗ *mangoldt* k / k) x − *Ln* x) ∈ $O(\lambda$-. *1*)
⟨*proof*⟩

4.4 The main result

We can now show the main result by extracting the primes we want using the orthogonality relation on characters, separating the principal part of the sum from the non-principal ones and then applying the above estimates.

lemma *Dirichlet-strong*:
  assumes *coprime h n*
  shows ($\lambda x.$ ($\sum p$ | *prime p* $\wedge$ [$p = h$] (*mod n*) $\wedge$ *real p* $\leq x$. *ln p* / *p*) $-$ *ln x* / *totient n*)
  $\in O(\lambda$-. *1*) (**is** ($\lambda x.$ *sum* - (*?A x*) $-$ -) $\in$ -)
⟨*proof*⟩

It is now obvious that the set of primes we are interested in is, in fact, infinite.

theorem *Dirichlet*:
  assumes *coprime h n*
  shows *infinite* {*p*. *prime p* $\wedge$ [$p = h$] (*mod n*)}
⟨*proof*⟩

In the future, one could extend this result to more precise estimates of the distribution of primes in arithmetic progressions in a similar way to the Prime Number Theorem.

end
end
end

References

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

[2] M. Carneiro. Formalization of the prime number theorem and dirichlet's theorem. In *Proceedings of the 9th Conference on Intelligent Computer Mathematics (CICM 2016)*, pages 10–13, 2016.

[3] J. Harrison. A formalized proof of Dirichlet's theorem on primes in arithmetic progression. *Journal of Formalized Reasoning*, 2(1):63–83, 2009.

[4] D. Newman. *Analytic Number Theory*. Number 177 in Graduate Texts in Mathematics. Springer, 1998.